The Journal of Systems and Software 140 (2018) 91–108

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

A truthful combinatorial double auction-based marketplace mechanism for cloud computing

Dinesh Kumar[a], Gaurav Baranwal[b,*], Zahid Raza[a], Deo Prakash Vidyarthi[a]

[a] School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India
[b] Department of Computer Science, Banaras Hindu University, Varanasi 221005, UP, India

ARTICLE INFO

Article history:
Received 8 November 2017
Revised 16 February 2018
Accepted 2 March 2018
Available online 6 March 2018

Keywords:
Cloud computing
Double auction mechanism
Resource allocation
Resource pricing
Truthfulness

ABSTRACT

Designing market-based mechanism that benefits both the cloud customer and cloud provider in a cloud market is a fundamental but complex problem. Double auction is one such mechanism to allocate resources that prevents monopoly and is used to design an unbiased optimal market strategy for cloud market. This work proposes a truthful combinatorial double auction mechanism for allocation and pricing of computing resources in cloud. For resource allocation, utilitarian social welfare maximization problem is formulated using Integer Linear Programming (ILP) and a near optimal solution is obtained using Linear Programming based padded method. For payment, truthful and novel schemes are designed for both customers and providers. Moreover, the proposed mechanism is individual rational, computationally tractable, weakly budget-balance and asymptotic efficient. Performance evaluation and comparative study exhibit that the proposed mechanism is effective on various performance metrics such as utilitarian social welfare, total utility, customers' satisfaction, providers' revenue and hence is applicable in real cloud environments.

© 2018 Elsevier Inc. All rights reserved.

1. Introduction

Cloud computing is a new computing based business model where various resources such as CPU, Network, Storage, Memory etc. are offered as utility and are available on demand (Buyya, 2009). Cloud service providers such as Amazon, Google, Microsoft etc. use different pricing schemes to attract the customers i.e. they want to increase their revenue. Cloud users want to use cloud services to execute their jobs or applications but by paying optimal price with desired QoS. Economics based approach such as auction, bargaining, distributive justice etc. have been widely used in various computing environments such as grid computing (Buyya et al., 2002; Li et al., 2009), cloud computing etc. (Baranwal et al., 2017; Baranwal and Vidyarthi, 2014; Kumar et al., 2017; Xu et al., 2011) to achieve objectives of providers and customers both.

Auctions, where price is determined by the supply and demand of the resources (Klemperer, 2004), are applications of mechanism design if one wants to design an auction with some desirable auction properties such as truthfulness, individual rationality, budget-balance etc. An auction mechanism basically consists of two parts: Allocation function and Payment scheme which need to be designed carefully to achieve auction properties. Recently, auction has been used for selling the underutilized and spare cloud resources (AWS, 2016). Auctions are decentralized, easy to implement and well suited for distributed systems like grid computing, cloud computing etc.

In past, researchers have proposed resource allocation models in cloud computing based on double auction. Most of the models mainly focused on the allocation schemes and little attention has been given to payment aspect. Some of the mechanisms have designed pricing schemes which are either truthful for one-side of market or not truthful at all. Moreover, to the authors' best knowledge, there is not a single combinatorial double auction mechanism in cloud which is budget-balanced and truthful for all participants. Keeping these issues in mind, this work proposes a truthful double auction for a combinatorial/multi-unit multi-item cloud market being referred as **T**ruthful **C**ombinatorial/ **M**ulti unit multi item **D**ouble **A**uction for **C**loud Computing (TCMDAC). In TCMDAC, each cloud user demands multiple types of Virtual Machines (VMs) in form of bundle as cloud users generally demand the resources in the form of bundle (Baranwal and Vidyarthi, 2015; Samimi et al., 2016) and each provider offers multiple units of multiple types of VMs. TCMDAC uses LP based padding method for Cloud computing environment where a provider offers multiple

* Corresponding author.
E-mail addresses: dineshyadav2571@gmail.com (D. Kumar), gaurav.vag@gmail.com (G. Baranwal), zahidraza@mail.jnu.ac.in (Z. Raza), dpv@mail.jnu.ac.in (D.P. Vidyarthi).

https://doi.org/10.1016/j.jss.2018.03.003
0164-1212/© 2018 Elsevier Inc. All rights reserved.

types of VMs. A Virtual Padding User (VPU) is considered which increases the competition among users and eliminates the users with less credentials (i.e. users with less budget (less bid value) or more required resources or both). It is assumed that VPU is having unlimited budget. TCMDAC exhibits various interesting features such as it supports the combinatorial bidding and it enables simple decision making to produce a near-optimal allocation. Truthful payment for all users is designed using critical payments while marginal cost is used to calculate the truthful payment for cloud providers. The key contributions of this work are as follows:

- To the best of authors' knowledge, TCMDAC is the first truthful combinatorial double auction for cloud market which is truthful for all participants (cloud customers as well as cloud providers) and weakly budget-balanced.
- In TCMDAC, a LP (Linear programming) based padding method is used. The generated allocations are near optimal, asymptotic efficient and can be computed in polynomial time.
- Novel payment schemes are designed for both customer and provider in a way to achieve truthfulness and budget-balance.
- It is shown theoretically as well as practically that TCMDAC is individual rational, incentive compatible, weakly budget-balanced and asymptotic efficient.
- TCMDAC is compared with state of the art and the experimental results show that it is effective, efficient and applicable in real cloud environments.

The outline of the paper is as follows. Section 2 gives an overview of related work on double auction mechanisms in cloud computing. Section 3 describes the system model and problem formulation. Section 4 describes the proposed TCMDAC model. Section 5 presents the performance evaluation and comparative study through simulation. Section 6 concludes the work with some possible future directions.

2. Related work

Auction, a market design mechanism, is very helpful for designing and modeling the competitive market (Klemperer, 2004). Its various variants like single sided (Zaman and Grosu, 2013), double sided (Baranwal and Vidyarthi, 2015; Kumar et al., 2017; Samimi et al., 2016), forward auction (Mashayekhy et al., 2015), reverse auction (Baranwal and Vidyarthi, 2016), first price auction, second price auction (Zaman and Grosu, 2012) etc. are quite useful in different market situations and have been used in cloud computing for resource allocation.

Double auction mechanisms, where bidding is done from both the market players i.e. customer and provider, provide a concrete and suitable framework for modeling the both side completion in auction based cloud market. In addition, use of double auction instead of repeated single-sided auction reduces the computational burden and complexity on the provider side (Wise and Morrison, 2000). One-sided auctions also reduce the possible trades or transactions, especially in combinatorial auctions (de Vries and Vohra, 2003; Rothkopf et al., 1998). Double auction is a many-to-many auction that prevents monopoly and can be used to design an unbiased optimal market strategy for a cloud market. It is proven that in double auction, efficiency maximizing mechanism yields more revenue compared to the single sided auction in the long run (Wise and Morrison, 2000). Therefore, various benefits of double auction e.g. dynamic pricing, efficient resource allocation, supply and demand principle, less time consumption and consideration of both side competitions make it suitable for the cloud computing market (Bratton et al., 1982; Cason and Friedman, 1996; Kumar et al., 2017).

Double auction based resource allocation and pricing mechanisms have been applied in grid computing before cloud computing (Grosu and Das, 2004; Izakian et al., 2010; Li et al., 2009). Li et al. (2009) considered combinatorial bidding and proposed combinatorial double auction based resource allocation and pricing schemes for the grid market. Although the work claims the incentive compatible property through experimental studies, but the work does not satisfy incentive compatible property theoretically. Grosu and Das (2004) used three most popular double auctions for resource allocation in grid: McAfee Double Auction (PMDA), Threshold price Double Auction Protocol (TPDA) and Continuous Double Auction (CDA). Grosu and Das (2004) shows that CDA performs better than PMDA and TPDA in terms of resource utilization. Motivated by the work proposed by Grosu and Das (2004), Izakian et al. (2010) proposed a continuous double auction based resource allocation for grid computing where grid users request for the resources in an auction market for executing their jobs. In Izakian et al. (2010), a user's bid value increases with the decrease in the number of remaining resources or average mean remaining time as it tries to finish its running tasks as soon as possible by acquiring more resources which can be obtained by bidding higher values. The provider's bid value is determined by the total workload and fluctuates between its ask price and maximum price. After that, trading price is determined by taking an average of highest bid and lowest ask price. Economic efficiency and System performance were two criteria which were used in Izakian et al. (2010). Simulation results prove that the model performs better in terms of fairness deviation, resource utilization and mean trade price.

A Combinatorial Double Auction based resource Allocation model named CDARA in cloud computing environment has been proposed in Samimi et al. (2016). The resource allocation has been done using greedy schemes which approximate the solution by a factor of $\sqrt{M}$ where M is the total resource quantity offered in the market. Average pricing mechanisms have been used for users and providers, previously used in Li et al. (2009). Two evaluation criterion: economic efficiency and incentive compatibility have been used in experimental studies. Though the model claimed to be truthful as Li et al. (2009) through experimental studies, it is not truthful theoretically. The reason is that average pricing mechanism used in Li et al. (2009) and Samimi et al. (2016) would leave the scope for users and providers to manipulate the cloud market by bidding falsely (Baranwal and Vidyarthi, 2015).

A Fair, Multi-attribute Combinatorial Double Auction Model (FMCDAM) for cloud environment is proposed in Baranwal and Vidyarthi (2015). In FMCDAM, various QoS attributes were considered along with price for winner determination and resource allocation is done using greedy technique as used in Samimi et al. (2016). FMCDAM reduces the bidder drop problem by allocating resources in a fair manner. Moreover, if a provider offers false QoS assurance then a penalty is imposed on the provider and its reputation is decreased which lowers its winning chances in successive rounds. However, the mechanism does not maximize the social welfare and is not truthful either.

One way to handle the wrong market manipulation is designing a truthful auction mechanism which gives incentives to participants for revealing their true information. Another way, to stop the malicious behavior of market participants, is feedback rating based reputation system as proposed in Sun et al. (2013) and Wang et al. (2015). In Wang et al. (2015) the Winner Determination Problem (WDP) problem is solved using Paddy Field Algorithm (PFA) whereas Sun et al. (2013) solves the WDP problem using Group Search Optimization Algorithm. A family of greedy based combinatorial double auction allocation mechanisms has been proposed in Chichin et al. (2015b). In this, the authors only designed the allocation mechanism without proposing any pricing mechanisms. Two types of sorting criteria are considered for homogeneous and heterogeneous resources in cloud. Resource Relative Relation (RRR) function and Resource Scarcity Factor (RSF) have been conceived as sorting criteria for homogeneous and heterogeneous resources respectively. According to RRR function, a trade is more efficient if it generates more surpluses for less number of traded resources. RSF considers the scarcity factor associated with the cloud resources. According to RSF, a trade is more efficient if it generates more surpluses for less number of traded scarce resources. All the sorting criteria are evaluated in terms of allocative efficiency, resource utilization and social welfare. Similar type of sorting criteria were also used in forward auction scenario for allocation of virtual machines among multiple users in Nejad et al. (2015).

Double auction mechanisms proposed in Chichin et al. (2015a) and Sun et al. (2015) ensure the truthfulness for cloud users only whereas mechanisms proposed in Wu et al. (2016) ensures the truthfulness for cloud providers only. Mechanisms proposed in Baranwal and Vidyarthi (2015), Chichin et al. (2015b), Kumar et al. (2017), Lee et al. (2015), Li et al. (2009) and Samimi et al. (2016) focus only on maximizing the social welfare whereas incentive compatibility has not been enforced for any participants. In some works, such as Sun et al. (2013) and Wang et al. (2015), rather than truthfulness a different approach has been adopted to stop malicious behavior of participants. To the author's best knowledge, a combinatorial double auction for cloud market has not been designed so far which is truthful for all the participants i.e. cloud users as well as cloud providers. The proposed model is budget-balanced, individual rational, truthful and asymptotic efficient.

3. System model and problem formulation

In this section, the system model of the proposed TCMDAC, its assumptions and problem formulation is discussed in detail.

3.1. Cloud market for TCMDAC

Combinatorial bidding is considered in this work which is quite reasonable in cloud market. It is because cloud resources (VMs) in cloud market are heterogeneous, substitute, complement and interrelated with each other. A user's jobs may have different resource requirements sometimes combination of VMs for their execution. Such type of requirements can be observed in web host applications (Huang et al., 2014; Urgaonkar et al., 2007). In another case, when a user builds workflow applications in a cloud computing environment, specifically on the Platform-as-a-service (PaaS), it needs to compose multiple types of services which are hosted on different types of VMs. In the proposed model, the single-minded cloud user is assumed as previously assumed in some other works also (Baranwal and Vidyarthi, 2016; Samimi et al., 2016). This means, if a user will get all types of VMs, only then it will be able to host its applications. Therefore, the users bid for the combination of VM instances. As the complete bundle could ensure the correct deployment of the application, no partial allocation is allowed in the whole allocation process. Table 1 depicts the list of symbols used in this work.

A VM is itself a bundle of resources possibly CPU, memory, storage, bandwidth etc. These resource attributes define or describe the characteristics of a VM. A cloud provider may provide various types of VMs differing in their resource configuration such as number of CPU cores, memory, available storage, OS type, geographical region etc. As an example, Microsoft Azure (Microsoft, 2016) and Amazon EC2 (AWS, 2016) provides different types of VMs. These VMs can be relatively compared in terms of their resource configuration by assigning a weight to each type of VMs. Suppose $\mathcal{K}$ is the set of all types of VMs available in the cloud market which is known to each market participant a-priori to auction. To represent the relative resource configuration of different type of VMs for comparing VMs of different types, a weight vector can be represented as $W = (w_1, w_2, \ldots, w_K)$ where w_k is any positive real number. It is assumed that there is a clear upward scaling (although not always proportional) from small instances to large instances i.e. $w_1 \le w_2 \le \cdots \ldots \le w_K$. The values of these weights are assumed to be known by every provider (CloudHarmony, 2010).

Cloud provider

Assume there are M cloud resource providers in the cloud market and $\mathcal{M} = 1, 2, \ldots, M$ where $\mathcal{M}$ denotes the set of cloud providers. These providers participate in the auction for providing cloud services to the cloud users. A provider's bid i.e. its offerings can be represented as a vector comprising its offered resource quantity and quoted ask price/cost for each type of resource. $Bid_j^p = (qp_j, pp_j)$ where qp_j is the resource vector that can be represented as: $qp_j = \langle qp_1^j, qp_2^j, \ldots, qp_K^j \rangle$. Here qp_k^j represent the number of VMs of type k offered/provided by provider j. The second vector in Bid profile of provider j is pp_j, which is the per-unit ask price/cost of each type of VM resources. pp_j is a cost vector and can also be represented as: $pp_j = \langle pp_1^j, pp_2^j, \ldots, pp_K^j \rangle$ where pp_k^j represents the ask price of a single VM of type k offered by the provider j.

All types of VMs offered by the cloud provider j are different and are scaled by weight vector W as described earlier. In the proposed model, the weights have been used to calculate the resources' pricing. As weights of VMs are clearly upward scaled, the per-unit prices of different types of VMs are considered according to their weight vector in such a way that $pp_1^j < pp_2^j < \ldots \cdots pp_K^j \; \forall \; j \in \mathcal{M}$. In addition, a new term called marginal cost have been coined that is used to rank all units of VMs in ascending order of their per-unit ask prices. Suppose $mc^k[q]$ represent the ask price of cheapest q-th unit of kth type of VM where q is the smallest integer greater than or equal to q. Suppose $\overline{pp_j}$ represents the actual cost vector of provider j. While bidding, a provider's bid pp_j may or may not be equal to its actual valuations $\overline{pp_j}$ e.g. a provider may falsely report its valuation (by bidding lower than its actual valuation) of resources in order to increase its chance of winning. It can also misrepresent its offered quantities and QoS. Normally, cloud resource provisioning and allocation happen over the internet where the cloud providers and users are remotely located. Thus, it is not possible to force providers or customers to bid truthfully (Bratton et al., 1982). In order to avoid wrong manipulation of whole market, proposed auction mechanism is designed in such a way that it provides incentives to providers to report its preferences truthfully.

Designing a truthful mechanism depends upon the nature of the bid attribute. A bid attribute can be categorized into verifiable and non-verifiable attributes (Pla et al., 2015) e.g. price is a non-verifiable attribute because it is known only to the provider and cannot be checked by any other participants due to its subjectivity. On the other hand, QoS and quantity of offered resources come into the category of verifiable attributes i.e. values of these attributes can be verified after allocation of resources. The mechanism proposed in this work provides the incentives to the providers to report only the ask prices truthfully. In case, if provider wins by falsely reporting the quantity offered, a penalty mechanism or a reputation mechanism can be applied which lower the chances of winning of that provider in the successive auction rounds (Baranwal and Vidyarthi, 2016, 2015; Ray et al., 2011). Therefore, the strategy space of a selfish user or provider is restricted for misstating the valuation about bundles i.e. for all bid profiles of provider p_j, $qp_j = \overline{qp_j}$ where $\overline{qp_j}$ is the actual resource quantity offered by provider j. In addition, the assumption regarding the assured QoS after winning the auction suits the reservation and on-demand type instances in which the number of times QoS violation are very less. In case of any QoS violation, a penalty

Table 1
Notations.

Notation	Description	Notation	Description
$\mathcal{N}$	Set of all cloud users	$\mathcal{M}$	Set of all cloud providers
N	Number of cloud users	M	Number of cloud providers
K	Number of VM types	$\mathcal{K}$	Set of VM types
$\overline{qu_i}$	Actual value of VM bundle requested by user i	qu^i_k	Number of kth type of VM requested by user i
$\overline{pu_i}$	Actual valuation of user i	pu_i	Quoted bid price by user i
$\overline{qp_j}$	Actual set of VMs supplied by provider j	qu_i	Quoted VM bundle requested by user i
qp_j	Quoted set of VMs supplied by provider j	qp^j_k	Number of kth type of VM offered by provider j
pp^j_k	Ask price of a single VM of type k offered by the provider j	pp_j	Quoted offer price by provider j.
mc	Marginal cost	$mc^k[q]$	cost of cheapest qth unit of kth type of VM
$mc^k_{-j}[q]$	cost of cheapest qth unit of kth type of VM without j provider's participation	x^{final}	Final user allocation vector
z^{final}	Final provider allocation matrix	pay^u	Users' payment vector
pay^p	Provider's payment vector	$\mathcal{U}^u_i$	Utility of user i
$\mathcal{U}^p_j$	Utility of provider j	$\mathcal{W}$	Social welfare
$VPU\ \mathcal{V}$	Virtual padding user	$qu_{\mathcal{V}}$	VM bundle of VPU $\mathcal{V}$
$qu^{\mathcal{V}}_k$	Number of kth type of VM requested by $\mathcal{V}$	$\bar{\mathcal{W}}$	Social welfare with $\mathcal{V}$
x'	User allocation vector with $\mathcal{V}$	z'	Provider allocation matrix with $\mathcal{V}$
$criticalValue^u_i$	Critical value of user i	$\mathcal{N}^s$	Set of winning users in x'
x''	Final user allocation vector	z''	Final provider allocation matrix
$\overline{pp_j}$	Actual cost/valuation of provider j		

mechanism can be imposed (Baranwal and Vidyarthi, 2015). It is also assumed that offered bid of a cloud provider is divisible i.e. it can provide its resources to multiple users simultaneously.

Cloud user

Suppose there are N cloud users in the cloud market and $\mathcal{N} = 1, 2, \ldots, N$ where $\mathcal{N}$ denote the set of all users. Depending upon its requirements, a user estimates the number of resources required and formulates its total demand in the form of bundles of virtual machines and estimates a bid price for the complete bundle of VMs. A user i bids for bundle $qu_i = < qu^i_1, qu^i_2, \ldots, qu^i_K >$ where qu^i_k represents the number of VM instances of type k requested by user i and $qu^i_k \geq 0$. In addition, user specifies a bid value pu_i for the requested bundle. For each bundle qu_i, the user has some valuation $\overline{pu_i}$ which is a function of qu_i and $\mathcal{P}$ i.e. $\overline{pu_i} = f(qu_i, \mathcal{P})$ where $\mathcal{P}$ is a set of factors that may be used while calculating the valuation such as budget, market price etc. The quoted bid price of the user i.e. pu_i may or may not be equal to his true valuation $\overline{pu_i}$. We assume that the "*request divisibility*" of cloud users is allowed i.e. a cloud user can acquire his resources from multiple providers. We also assume that a user can be untruthful in terms of his bid price only, not in terms of required quantity i.e. $qu_i = \overline{qu_i}$ where $\overline{qu_i}$ is his actual required resource quantity. The reason for the above assumption is that misreporting the required bid quantity always results in zero utility for a user because in this case user would not get the complete bundle if it wins.

Cloud auctioneer

Cloud Auctioneer is an entity (an individual, business organization or a broker firm) who holds the auction for trading of resources in a cloud market. Auctioneer possesses all the technical details of the market and its components such as cloud resource configurations, nature of traded resources, total number of users and providers in the cloud market.

For simplicity, in this paper cloud Service User, cloud Service Provider and cloud Auctioneer will be referred as user, provider and auctioneer respectively

3.2. Problem formulation

Since aim of this work is to benefit both the users and providers, the goal of the resource allocation problem is to maximize total social welfare i.e. the difference between the users' total payment and providers' total revenue while satisfying the resource availability constraint. If all users and providers bid truthfully, the social welfare maximization problem $\Pi(\mathcal{N}, \mathcal{M})$ can be formulated as an Integer Linear Programming (ILP) problem which is an NP-Hard Problem.

$$\Pi(\mathcal{N}, \mathcal{M}) : maximize\ \mathcal{W}(\mathcal{N}, \mathcal{M}) = \sum_{i \in \mathcal{N}} pu_i x_i - \sum_{j \in \mathcal{M}} \sum_{k \in \mathcal{K}} pp^j_k z_{jk} \tag{1}$$

Subject to

$$\sum_{i \in \mathcal{N}} qu^i_k x_i = \sum_{j \in \mathcal{M}} z_{jk}\ \forall k \in \mathcal{K} \tag{2}$$

$$x_i \in \{0, 1\}\ \forall\ i \in \mathcal{N} \tag{3}$$

$$z_{jk} \in \left\{0, 1, \ .. \ ., \ qp^j_k\right\}\ \forall\ j \in \mathcal{M},\ k \in \mathcal{K} \tag{4}$$

In the above formulation, in Eq. (1) $\mathcal{W}(\mathcal{N}, \mathcal{M})$ i.e. social welfare is the objective to be maximized. First constraint, i.e. Eq. (2), specifies that the number of requested resource should be equal to number of offered resources in the final allocation. Eqs. (3) and (4) depict that decision variables x_i and z_{jk} should be integers. x_i is 1 if user i wins otherwise 0. z_{jk} denotes the allocated quantity of kth type of VM of provider j and it should not exceed the total number of VMs of kth type at provider j.

3.2.1. Designing target and objectives

In complex scheduling situations, such as combinatorial double auctions where bidders bid in the form of bundles, economic efficiency and computational efficiency conflict with each other (Xia et al., 2005). Also, a double auction mechanism can't be economically efficient and truthful at the same time (McAfee, 1992). Among all the above properties, two properties i.e. individual rational and budget balance are necessary for sustainable auctions i.e. bidders will not take part in auction voluntarily if they incur loss by participating in the auction and the auctioneer will not perform auction in long run if the mechanism does not satisfy budget balance property. The above problem is solved keeping in mind the economic properties one needs to satisfy.

Asymptotic efficiency

Economic efficiency of a double auction mechanism is measured in the terms of total social welfare generated by the mechanism. An efficient mechanism maximizes the total social welfare. A double auction mechanism is 100% efficient if the actual welfare generated by the mechanism is equal to the welfare generated theoretically i.e. there is no loss of welfare in the mechanism.

Another term i.e. asymptotic efficiency is a weaker notion of efficiency. A mechanism is said to be asymptotic efficient when welfare loss converges to zero as maximal social welfare approaches infinity (Xia et al., 2005).

Incentive compatibility

Incentive compatibility or truthfulness ensures that bidding truthfully (revealing true information) is a dominant strategy for each participant i.e. for any user i, $\mathcal{U}_i^u(\overline{pu_i}) \geq \mathcal{U}_i^u(pu_i)\ \forall\ pu_i$ and for any provider j, $\mathcal{U}_j^p(\overline{pp_j}, \overline{qp_j}) \geq \mathcal{U}_j^p(pp_j, qp_j)\ \forall\ pp_j, qp_j$ where $\mathcal{U}_i^u$ is utility of user i and $\mathcal{U}_j^p$ is utility of provider j.

Budget-balance

Budget-balance property ensures that the total payment done by users should be equal to the total payment received by the providers. i.e. $\sum_{i\in\mathcal{N}} pay^i = \sum_{j\in\mathcal{M}} pay^j$ where pay^i is payment paid by user i and pay^j is payment received by provider j. If $\sum_{i\in\mathcal{N}} pay^i - \sum_{j\in\mathcal{M}} pay^j \geq 0$, then the mechanism is said to be weak budget-balance.

Individual rational

A mechanism is said to be individual rational if utility of all participants is always non-negative i.e. a user doesn't pay more than the bid valuation and a provider doesn't get payments less than its actual valuation (reservation, ask price or cost) i.e. $\mathcal{U}_i^u \geq 0$ and $\mathcal{U}_j^p \geq 0\ \forall\ i \in \mathcal{N}$ and $\forall\ j \in \mathcal{M}$

Computational efficiency

A double auction mechanism is said to be computationally tractable if the allocation and payments can be calculated in polynomial time.

4. TCMDAC: the proposal

The proposed TCMDAC model contains several phases detailed as below.

Start of auction

Cloud auctioneer starts the auction by inviting users and providers to submit their bids.

Submission of bids

All users and providers, participating in double auction, report to the auctioneer with their bids. Each user bids the combinatorial requests whereas each provider advertises its resources coupled with per-unit ask prices. After submission, auctioneer closes the auction and calculates the allocation and payment output vector for both users and provider.

Resource allocation

As discussed earlier, designing an efficient, individual rational, truthful and budget-balance double auction mechanism is impossible even in simple environments (McAfee, 1992; Myerson, 1981). Some works such as Chu and Shen (2008, 2006, 2007) and Huang et al. (2002) designed truthful double auction mechanisms for combinatorial/single-unit environment in which each user has combinatorial requests for resources and each seller offers single unit of a particular type of commodity. If we extend these mechanisms for multi-unit settings in cloud, then the mechanism loses budget-balance property and incur a budget-deficit (Chu, 2009). A double auction mechanism for bundle/multi-unit environments is proposed in Chu (2009) by designing a padding method to recover the budget-deficit by inserting a gap between the resource requirements and resource offers. This gap is filled with the help of a virtual user who has unlimited budget and large resource requirements. According to supply-demand principle, consideration of Virtual Padding User (VPU) increases the total resource requirements of users which results in higher equilibrium price, higher buying price, less traded quantities and low selling price thus generating budget-surplus (Chu, 2009).

Motivated by Chu (2009) which considers the multi-unit environment for a single type of commodity only, this work uses padding method but for cloud computing environment where a provider offers multiple types of VMs. A VPU $\mathcal{V}$ is considered which increases the competition among users and eliminates the users with less credentials (i.e. users with less budget (less bid value) or more required resources or both). It is assumed that VPU is having unlimited budget. The requested quantities by VPU is set as the maximum quantity of each type of VM offered by any provider i.e. $qu_{\mathcal{V}} = qu_1^{\mathcal{V}}, qu_2^{\mathcal{V}}, \ldots, qu_K^{\mathcal{V}}$ where $qu_k^{\mathcal{V}} = \max\ (qp_k^j)\ \forall\ k \in \mathcal{K},\ j \in \mathcal{M}$. The intuition behind the use of VPU $\mathcal{V}$ with such requirements is that normally, a provider with the largest supply has the largest power in manipulating the prices. But large size of $\mathcal{V}$ may also result in large efficiency loss. Therefore $\mathcal{V}$ with suitable resource demands is designed. In Theorem 9, it is shown that for each type of resources, the above designed VPU helps in designing the mechanism as incentive compatible, individual rational and budget-balance. The social welfare maximization problem with consideration of VPU $\mathcal{V}$ is $\bar{\Pi}(\mathcal{N}, \mathcal{M}, \mathcal{V})$ which can be depicted as below.

$$\bar{\Pi}(\mathcal{N}, \mathcal{M},\ \mathcal{V}) : maximize\ \bar{\mathcal{W}}(\mathcal{N}, \mathcal{M}, \mathcal{V})$$
$$= \sum_{i\in\mathcal{N}} pu_i x_i - \sum_{j\in\mathcal{M}} \sum_{k\in\mathcal{K}} pp_{jk} z_{jk} \tag{5}$$

Subject to:

$$\sum_{i\in\mathcal{N}} qu_k^i x_i + qu_k^{\mathcal{V}} = \sum_{j\in\mathcal{M}} z_{jk}\ \forall k \in \mathcal{K} \tag{6}$$

$$0 \leq x_i \leq 1\ \forall i \in \mathcal{N} \tag{7}$$

$$0 \leq z_{jk} \leq qp_k^j\ \forall\ j \in \mathcal{M},\ k \in \mathcal{K} \tag{8}$$

The difference between the $\Pi(\mathcal{N}, \mathcal{M})$ and $\bar{\Pi}(\mathcal{N}, \mathcal{M}, \mathcal{V})$ is the consideration of VPU $\mathcal{V}$ and relaxation of some constraints i.e. values of decision variables x_i and z_{jk} can be fractional rather than integer. Let the solution generated by solving the $\bar{\Pi}(\mathcal{N}, \mathcal{M},\ \mathcal{V})$ problem is (x', z'). After finding the solution (x', z'), critical values of all users are calculated. Here, critical values for a user i can be defined using equation below.

$$criticalValue_i^u = \inf\{qp_i |\ x_i' = 1\} \tag{9}$$

Critical value for a user is equal to the minimum value, it can bid while remaining in solution (x', z') with $x_i' = 1$ for padded optimization problem $\bar{\Pi}(\mathcal{N}, \mathcal{M},\ \mathcal{V})$, given the bids of others are unchanged. To find the critical value, a binary search over a defined range of values is performed. As the mechanism have to be individual rational, a user payment cannot be more than its actual valuation i.e. $criticalValue_i^u \leq \overline{pu_i}\ \ \forall i \in \mathcal{N}$. Therefore, upper bound for the critical value is its actual valuation. Lower bound of critical value is established using Theorem 6 i.e. $criticalValue_i^u \geq \sum_{k\in\mathcal{K}} mc^k[\sum_{u\in\mathcal{N}} qu_k^u x_u' + qu_k^{\mathcal{V}}\ ] * (qu_k^i)$. CRITICAL-VALUE function, given in Algorithm 3, lists the procedure to calculate the critical values for all users.

After finding the critical values for all cloud users, a new set of winning users $\mathcal{N}^s$ is constructed where a user $i \in \mathcal{N}^s$ if $x_i' = 1$. The new set of winning users $\mathcal{N}^s$ contains those users who bid higher than their critical values. The interesting point here is that users in $\mathcal{N}^s$ are more competitive and eligible as compared to the losing users. As VPU has the largest valuation among all competing users, it always wins in padded optimization problem $\bar{\Pi}(\mathcal{N}, \mathcal{M},\ \mathcal{V})$. This causes some less eligible users (less valuation or more requested quantity or both) to lose who may winners without considering VPU. After finding the new winning set, the set of winning users are allowed to trade with all providers without considering VPU $\mathcal{V}$. The problem again can be formulated as a linear programming

Algorithm 1
TCMDAC-ALLOC.

Input: $Bid_i^u = (qu_i, pu_i) \forall i \in \mathcal{N}$; vector of users' requests (resource bundle, valuation)
Input: $Bid_i^p = (qp_i, pp_i) \forall i \in \mathcal{N}$; vector of providers' offers (resource bundle, valuation)
Input: Marginal cost $mc^k\ \forall k \in \mathcal{K}$
consider a VPU $\mathcal{V}$ with $qu_{\mathcal{V}} = qu_1^{\mathcal{V}}, qu_2^{\mathcal{V}}, \ldots, qu_K^{\mathcal{V}}$ Where $qu_k^{\mathcal{V}} = \max\ (qp_k^j)\ \forall\ k \in \mathcal{K},\ j \in \mathcal{M}$
reformulate the problem $\Pi(\mathcal{N}, \mathcal{M})$ into $\bar{\Pi}(\mathcal{N}, \mathcal{M}, \mathcal{V})$ using Eqs. (5)–(8).
solve $\bar{\Pi}(\mathcal{N}, \mathcal{M}, \mathcal{V})$ and the obtained solution is (x', z')
$criticalValue^u = CRITICAL\text{-}VALUE(Bid_i^u, (x', z'), qu_{\mathcal{V}}, mc^k)$
initialize $\mathcal{N}^s = \phi$.
for $i = 1$ to N
  if $pu_i > criticalValue_i^u$
    $\mathcal{N}^s = \mathcal{N}^s \cup \{i\}$
re-allocate with new set of user in $\mathcal{N}^s$ by solving $\bar{\Pi}(\mathcal{N}^s, \mathcal{M})$. Let the resultant solution is (x'', z'')
for $i \in \mathcal{N}$
  if $x_i' = x_i'' = 1$
    $x_i^{final} = 1$
  else
    $x_i^{final} = 0$
$z^{final} = z''$
OUTPUT: Final Allocation output for user and provider (x^{final}, z^{final})

based optimization problem $\bar{\Pi}(\mathcal{N}^s, \mathcal{M})$ as below.

$$\bar{\Pi}(\mathcal{N}^s, \mathcal{M}) : maximize\ \bar{\mathcal{W}}(\mathcal{N}^s, \mathcal{M}) = \sum_{i \in \mathcal{N}^s} pu_i x_i - \sum_{j \in \mathcal{M}} \sum_{k \in \mathcal{K}} pp_{jk} z_{jk} \tag{10}$$

Subject to

$$\sum_{i \in \mathcal{N}^s} qu_k^i x_i = \sum_{j \in \mathcal{M}} z_{jk}\ \forall k \in \mathcal{K} \tag{11}$$

$$0 \le x_i \le 1\ \forall i \in \mathcal{N}^s \tag{12}$$

$$0 \le z_{jk} \le qp_k^j\ \forall\ j \in \mathcal{M},\ k \in \mathcal{K} \tag{13}$$

Let the solution generated after solving the above problem is (x'', z''). If $x_i' = x_i'' = 1$ then $x_i^{final} = 1$, otherwise $x_i^{final} = 0$. Here, it can be shown that if $x_i' = 1$, then $x_i'' = 1$ will also be true. This can be justified by the fact that removal of VPU $\mathcal{V}$ from the users' set increases the chance of winning of users who are already winners. As VPU $\mathcal{V}$ has the largest demand, its removal from the users' set leave enough resource quantity for the remaining users. Here z'' is a $M \times K$ matrix where z_{jk}'' represent the number of VMs of type k allotted or offered by provider j. The whole allocation is presented in Algorithm 1.

Designing of pricing scheme

After allocation phase, trading prices are calculated by designing separate payment schemes for both users and providers. The payment schemes are different for users and providers because they have separate bidding configuration and allocation. The pricing schemes are designed for both user and provider in such a way that the mechanism is truthful, individual rational and budget-balance. TCMDAC-PAY function gives the payment output vector pay^u and pay^p where $pay^u = \{pay_1^u, pay_2^u, \ldots, pay_N^u\}$ and $pay^p = \{pay_1^p, pay_2^p, \ldots, pay_M^p\}$.

Cloud user's payment scheme

Critical value schemes have been designed to determine the payments of cloud users. In critical payment method, a user's payment doesn't depend upon its own bid rather on the bids of other users. If a user loses, it pays zero. If a user wins, it always pays equal to the critical value below which its bid have lost (Lehmann and O' Callaghan, 2002). Equation below describes the payment mechanism for cloud user.

$$pay_i^u = \begin{Bmatrix} criticalValue_i^u,\ x_i^{final} = 1 \\ 0,\ otherwise \end{Bmatrix}\ \forall\ i \in \mathcal{N} \tag{14}$$

Cloud provider's payment scheme

Critical payments method applied in cloud users' case is not applicable to the case of cloud provider. The reason is that critical payments require that the participant should be single minded i.e. a provider would like to allocate its resources fully or partially if it wins in the auction (Lehmann and O' Callaghan, 2002), while cloud provider's offers are divisible. The payments for cloud providers are calculated using marginal values of resources offered by various providers. The idea is motivated by the work proposed in Loertscher and Mezzetti (2013) which considers the concept of marginal value to provide truthful double auction for a market but in Loertscher and Mezzetti (2013) both buyer and seller trades multiple unit of a homogeneous good unlike our work. The payment of provider j can be calculated as given in equation below.

$$pay_j^p = \sum_{k \in \mathcal{K}} pay_{jk}^p = \sum_{k \in \mathcal{K}} \sum_{t=1}^{z_{jk}} mc_{-j}^k [1 + S_k - t] \tag{15}$$

where $S_k = \sum_{j \in \mathcal{M}} z_{jk}$ is the total number of VMs of type k allotted to the winning users in allocation which is equal to $\sum_{i \in \mathcal{N}^s} qu_k^i x_i''$ where x_i'' is final user allocation vector.

As the allocation maximizes the total surplus, providers bid are sorted in ascending order according to their offered prices. For a winning provider p_j, payment is calculated using marginal cost functions mc and mc_{-j} where mc_{-j} represents the marginal cost vector of VMs when provider j is not participating. If provider j wins, the offered price or cost of provider j is less than the cost of non-winning providers as providers are arranged in increasing order of their cost/ask prices. Therefore, when provider j is not participating, this results in the winning of one or more than one providers with higher bid prices (as providers are sorted in ascending order and provider j has lower offer/ask price than new winning providers) who successfully allocate their VMs. Moreover, provider j will get paid the offered price of new winning providers in order to achieve individual rationality and incentive compatibility. In other words, each provider gets the amount equal to the harm caused by him to other providers which is measured by the marginal cost of providing the resources by new winning providers. For example, it gets paid the S_k-th lowest marginal value of its competitors for the first unit of kth type of VM, $(S_k - 1)$-th lowest for the second unit and lowest for S_k-th unit where S_k is the total traded VMs of type k. The payment scheme for the same has been described in Algorithm 2.

Algorithm 2
TCMDAC-PAY.

Input: *Allocation Output* = (x^{final}, z^{final})
Input: *Critical value vector* = $criticalValue^u$
Input: *Marginal cost* $mc^k\ \forall k \in \mathcal{K}$
//user's truthful payments using critical value pricing mechanism
for each $i \in \mathcal{N}$
  if $x_i^{final} = 1$
    $pay_i^u = criticalValue_i^u$
  else
    $pay_i^u = 0$
//provider truthful payment mechanism based on marginal values
for each $k \in \mathcal{K}$
$S_k = \sum_{j\in\mathcal{M}} z_{jk}$
for each provider $j \in \mathcal{M}$
  if $\sum_{k\in\mathcal{K}} z_{jk}^{final} \neq 0$
    $pay_j^p = \sum_{k\in\mathcal{K}} pay_{jk}^p = \sum_{k\in\mathcal{K}} \sum_{t=1}^{z_{jk}} mc_{-j}^k [1 + S_k - t]$
  else
    $pay_j^p = 0$
OUTPUT: *Payment output vector for users and providers* (pay^u, pay^p).

Algorithm 3
CRITICAL-PAY.

Input: $Bid_i^u = (qu_i, pu_i)\forall i \in \mathcal{N}$; *vector of requests (resource bundle, valuation)*
Input: (x', z'); *Solution of padded optimization problem* $\bar{\Pi}(\mathcal{N}, \mathcal{M}, \mathcal{V})$
Input: $qu_{\mathcal{V}}$; *VPU* $\mathcal{V}$ *resource quantity*
Input: *Marginal cost* $mc^k\ \forall k \in \mathcal{K}$
for all $i \in \mathcal{N}$ *do*
  $lb = \sum_{k\in\mathcal{K}} mc^k [\sum_{u\in\mathcal{N}} qu_k^u x_u' + qu_k^{\mathcal{V}}] * (qu_k^i)$
  $ub = \overline{pu_i}$
  while $(ub - lb) \geq 1$ *do*
    $mid = \lceil (lb + ub) \rceil / 2$
    $Bid_i^u = (qu_i, mid)$
    find (x', z') *by solving* $\bar{\Pi}(\mathcal{N}, \mathcal{M}, \mathcal{V})$ *with updated user bid*
    if $x_i' = 1$ *then*
      $ub = mid$
      if $lb = ub - 1$ *then*
        $criticalValue_i^u = mid$
        break
    else
      $lb = mid$
      if $lb = ub - 1$ *then*
        $criticalValue_i^u = mid + 1$
        break
Ouput: $criticalValue^u = (criticalValue_1^u, criticalValue_2^u, , criticalValue_N^u)$

Payments to cloud users and providers

Allocation and payment functions' generated outcome is (x, pay^u) for user and (z, pay^p) for provider. After the final allocation and payments using TCMDAC-ALLOC and TCMDAC-PAY, the cloud users pay to the cloud auctioneer. After that, cloud auctioneer pays to the cloud providers as per the prices calculated using TCMDAC-PAY.

End of auction

The whole round of auction ends after complete payment process.

Utility of user i can be calculated using equation below.

$$\mathcal{U}_i^u = \overline{pu_i} x_i - pay_i^u \tag{16}$$

As a cloud user is single-minded, it will pay only if it gets its requested complete bundle, otherwise its payment will be zero. Similarly, utility of a provider will be derived from its payment and the quantity of resources offered in final allocation. The utility of provider j can be calculated using equation below.

$$\mathcal{U}_j^p = pay_j^p - \sum_{k\in\mathcal{K}} \overline{pp_k^j} z_{jk} \tag{17}$$

Properties of TCMDAC

Theorem 1. *TCMDAC is truthful for all cloud users.*

Proof. The proof of truthfulness for all the participants is derived from Vickrey (1961), Clarke (1971) and Groves's (1973) arguments. If user does not win, it pays zero and its utility will be zero, otherwise it pays the critical payment which is independent of its bid value pu_i and depends on the valuations of its competitors i.e. valuation of other users. This critical payment is equal to the minimum bid value it can bid in order to win in the auction. If user i bids higher than $criticalValue_i^u$, then it successfully obtains its resource bundle in the optimal solution $\bar{\Pi}(\mathcal{N}^s, \mathcal{M})$. If it bids lower than this critical payment, it loses in the auction, $i \notin \mathcal{N}^s$ (Lehmann and O'Callaghan, 2002). This critical payment method ensures incentive compatibility as proven in Lehmann and O'Callaghan (2002). In order to prove the mechanism truthful for users, we consider different use cases and proved that bidding truthfully is always in the best interest of user u_i i.e. user's utility will be maximum when it bids its actual valuation.

(a) If cloud user wins i.e. $x_i^{final} = 1$, then there are two possible cases:
  1. If $pu_i < \overline{pu_i}$ i.e. user underbids its actual valuation, then there are two further cases:
     - If $pu_i \geq criticalValue_i^u$, then user wins and acquires its VM resource bundle in an optimal solution (x', z') to $\bar{\Pi}(\mathcal{N}, \mathcal{M}, \mathcal{V})$. In that case $x' = 1$ and enters into set $\mathcal{N}^s$. Accordingly, it also wins in the final allocation i.e. $x'' = 1$. In this case total traded quantity of each type of VM is $\sum_{j\in\mathcal{M}} z_{jk} = \sum_{i\in\mathcal{N}} qu_k^i x_i' + qu_k^{\mathcal{V}}\ \forall\ k \in \mathcal{K}$ and lowest $\lceil \sum_{i\in N} qu_k^i x'_i + qu_k^V \rceil - $th offer price determines the marginal price of kth type of VM where $\lceil x \rceil$ is the smallest integer greater than or equal to x. The users in set $\mathcal{N}^s$ will now trade with the providers by solving $\bar{\Pi}(\mathcal{N}^s, \mathcal{M})$ which generates the final allocation (x'', z''). In the final allocation, a total $\sum_{i\in\mathcal{N}^s} qu_k^i x_i''$ VMs of type k are allocated to users which is equal to $\sum_{j\in\mathcal{M}} z_{jk}''$. At this point, marginal cost price of kth type of VM is the lowest $\sum_{i\in\mathcal{N}^s} qu_k^i x_i'' - th$ offer price which is significantly lower than the marginal prices to the optimal solution $\bar{\Pi}(\mathcal{N}, \mathcal{M}, \mathcal{V})$ because $\sum_{i\in\mathcal{N}} qu_k^i x_i' + qu_k^{\mathcal{V}} \geq \sum_{i\in\mathcal{N}^s} qu_k^i \geq \sum_{i\in\mathcal{N}^s} qu_k^i x_i''$ and also $mc^k [\sum_{i\in\mathcal{N}} qu_k^i x_i' + qu_k^{\mathcal{V}}] \geq mc^k \sum_{i\in\mathcal{N}^s} qu_k^i \geq mc^k \sum_{i\in\mathcal{N}^s} qu_k^i x_i''$. The above inequality ensures that if the user wins in the optimal solution of $\bar{\Pi}(\mathcal{N}, \mathcal{M}, \mathcal{V})$, it also wins in the final allocation to problem $\bar{\Pi}(\mathcal{N}^s, \mathcal{M})$. Therefore, when user bids higher than its critical value $criticalValue_i^u$, it acquires its bundle in $\bar{\Pi}(\mathcal{N}, \mathcal{M}, \mathcal{V})$ and also in $\bar{\Pi}(\mathcal{N}^s, \mathcal{M})$ at price $criticalValue_i^u$. In this case, the utility will remain the same.
     - If $pu_i < criticalValue_i^u$, user will lose if it bids lower than its critical payment i.e. $x' = 0$ and is not able to win in the final allocation. This results in zero utility for the user. Therefore, in this case, false bidding results in a decrease in its utility.
  2. If $pu_i > \overline{pu_i}$, then it will be always be a winner because it further increases the chance of winning it as allocation algorithm chooses the user with higher valuation to maximize the social welfare. This bidding case also results in no change in its utility as it is paying the same price as when it bids $pu_i = \overline{pu_i}$. This is because its payment that is equal to critical payment calculated using Eq. (9) depends upon the bid values of its competitors i.e. other cloud users.

(b) Cloud user loses i.e. $x_i^{final} = 0$, then there are two cases:

1. If $pu_i < \overline{pu_i}$, then it will still be a loser in the auction because lowering the valuation further lowers the chance of winning in the auction. This is due to the LP based allocation algorithm design as it will select only those users who have high valuation as it is designed to maximize the total social welfare (sum of valuations). Therefore, its utility will remain zero. In this case also, false bidding brings no change in its utility.
2. If $pu_i > \overline{pu_i}$, then there are two further cases:
   - It may still be the loser when it does not qualify the winning bid's amount i.e. higher than the bids of other users. Then, its utility will remain zero and there is no change in its utility by bidding falsely.
   - Suppose the user wins by bidding greater than its actual valuation and pays the critical payment which will be greater than its valuation $\overline{pu_i}$. But in that case it will pay more than its valuation which brings negative utility for him. Therefore, this case represents a reduction in its utility by bidding falsely.

All the above cases prove that if the user bids different than its actual valuations (whether greater than or less than), then the utility of the user either remains the same or is decreased. If user's valuation is higher than its critical payment, it prefers to win and gets its requested resources (trade), which can be achieved by bidding truthfully. If its actual valuation is lower than its critical payment, it prefers not to trade which can also be achieved by reporting its valuation truthfully and when its actual valuation equals its critical payment, it is indifferent to trading. This proves that bidding truthfully is always the users' dominant strategy and the proposed mechanism is incentive compatible for all cloud users. □

Theorem 2. *TCMDAC is individual rational for cloud user.*

Proof. In TCMDAC, the payments of cloud users depend upon the critical value pricing. If a user wins, it will pay its critical payment which is always less than its valuation. If it doesn't win, it pays zero and its utility will be zero. Therefore, in each case, a user's utility will be always greater than or equal to zero i.e. $\mathcal{U}_i^u \geq 0$. Therefore, the proposed mechanism is individual rational for all cloud users. □

Theorem 3. *The proposed mechanism in TCMDAC induces a feasible allocation in which each cloud user either acquire whole requested resource bundle or nothing and each cloud provider offers discrete units of VM resources.*

Proof. In the proposed payment mechanism, if $i \in \mathcal{N}^s$,user i wins in auction and successfully acquires its resource bundle in the solution (x', z') to problem $\bar{\Pi}(\mathcal{N}^s, \mathcal{M})$. If $i \notin \mathcal{N}^s$, user i loses and it gets nothing. Also, the total number of VMs allotted in the final allocation is $\sum_{i \in \mathcal{N}^s} qu_k^i x_i^{''} = \sum_{j \in \mathcal{M}} z_{jk}$ for $k \in \mathcal{K}$ which is definitely an integer value. As all providers' resources are ranked according to their offered price from low to high, each provider also offers discrete units of resource quantities. Therefore, the allocation generated by TCMDAC mechanism is feasible. □

Before proving the truthfulness for providers, first we will prove the following lemma

Lemma 1. *For any user $i \in \mathcal{N}^s$, $x_i^p = 1$ in the optimal solution (x^p, z^p) to $\bar{\Pi}(\mathcal{N}^s, \mathcal{M}\backslash\{j\})$ for any $j \in \mathcal{M}$ if $qu_k^{\mathcal{V}} > max_{j \in \mathcal{M}}\{qp_k^j\} - 1$, $\forall\ k \in \mathcal{K}$.*

Proof. At the optimal solution of (x', z') to $\bar{\Pi}(\mathcal{N}^s, \mathcal{M}, \mathcal{V})$, for each $k \in \mathcal{K}$, a total of $\sum_{i \in \mathcal{N}^s} qu_k^i x_i' + qu_k^{\mathcal{V}} = \sum_{j \in \mathcal{M}} z_{jk}$ units of VMs of type k are allocated to winning users and the highest ask price among the traded units is $mc^k[\sum_{i \in \mathcal{N}} qu_k^i x_i' + qu_k^{\mathcal{V}}]$. If $i \in \mathcal{N}^s$, $x_i^p = 1$. If we compare the solution to $\bar{\Pi}(\mathcal{N}, \mathcal{M}, \mathcal{V})$ at $x_i^p = 1$ and $x_i^p = 1 - \epsilon$ for small $\epsilon > 0$, then we have $pu_i \geq \sum_{k \in \mathcal{K}} qu_k^i * mc^k[\sum_{i \in \mathcal{N}} qu_k^i x_i' + qu_k^{\mathcal{V}}]$ for all feasible allocations. Also, we have assumed that VPU $\mathcal{V}$ has the largest demand i.e. $qu_k^{\mathcal{V}} = \max\ (qp_k^j)\ \forall\ k \in \mathcal{K},\ j \in \mathcal{M}$ and $qu_k^{\mathcal{V}} \geq qp_k^j\ \forall\ k \in \mathcal{K},\ j \in \mathcal{M}$, and $mc^k[\sum_{i \in \mathcal{N}} qu_k^i x_i' + qu_k^{\mathcal{V}}] \geq mc^k[\sum_{i \in \mathcal{N}} qu_k^i x_i' + qp_k^j]\ \forall\ k \in \mathcal{K},\ j \in \mathcal{M}$ and $mc^k[\sum_{i \in \mathcal{N}} qu_k^i x_i' + qu_k^{\mathcal{V}}] \geq mc^k[\sum_{i \in \mathcal{N}^s} qu_k^i + qp_k^j]\ \ \forall\ \ k \in \mathcal{K},\ \ j \in \mathcal{M}$. In addition, we have an another inequality which is always true i.e. $mc^k[\sum_{i \in \mathcal{N}^s} qu_k^i x_i^p + qu_k^{\mathcal{V}}] \geq mc_{-j}^k[\sum_{i \in \mathcal{N}^s} qu_k^i x_i^p]\ \ \forall j \in \mathcal{M}$ when $qu_k^{\mathcal{V}} > 0\ \ \forall\ \ k \in \mathcal{K}$. Also, we have $\sum_{i \in \mathcal{N}^s} qu_k^i \geq \sum_{i \in \mathcal{N}^s \backslash \{u\}} qu_k^i x_i^p + qu_k^u \geq \sum_{i \in \mathcal{N}^s} qu_k^i x_i^p$, Thus we have the following inequalities:

$$
\begin{aligned}
pu_i &\geq \sum_{k \in \mathcal{K}} qu_k^i * mc^k \left[\sum_{i \in \mathcal{N}^s} qu_k^i + qp_k^j \right] \\
&\geq \sum_{k \in \mathcal{K}} qu_k^i * mc^k \left[\sum_{i \in \mathcal{N}^s \backslash \{u\}} qu_k^i x_i^p + qu_i^u + qp_k^j \right] \\
&\geq \sum_{k \in \mathcal{K}} qu_k^i * mc^k \left[\sum_{i \in \mathcal{N}^s} qu_k^i x_i^p + qp_k^j \right] \\
&\geq \sum_{k \in \mathcal{K}} qu_k^i * mc_{-j}^k \left[\sum_{i \in \mathcal{N}^s} qu_k^i x_i^p \right]
\end{aligned}
$$

In $\bar{\Pi}(\mathcal{N}^s, \mathcal{M}\backslash\{j\})$, if $x_i^p < 1$, we could always increase x_i^p to 1 to achieve a higher social welfare. Therefore, $x_i^p = 1$ must be in the optimal solution of $\bar{\Pi}(\mathcal{N}^s, \mathcal{M}\backslash\{j\})$. □

Theorem 4. *In TCMDAC, bidding truthfully is dominant strategy for each provider and non-uniform pricing scheme is forced on each provider.*

Proof. In the proposed mechanism, all the providers, who successfully allocate their resources to cloud users, get paid from the auctioneer/market maker. For each provider, if a provider wins, it will get paid by the auctioneer the cost equal to the cost of harming other providers. For simplicity, we prove the truthfulness property for any VM type k which can be generalized for all $k \in \mathcal{K}$. We will consider all the cases and prove that revealing true offers cost/ask prices is the dominant strategy. As explained in Eq. (17), utility of a provider j is as: $\mathcal{U}_j^p = pay_j^p - \sum_{k \in \mathcal{K}} \overline{pp_k^j} z_{jk}$, where $pay_j^p = \sum_{k \in \mathcal{K}} pay_{jk}^p = \sum_{k \in \mathcal{K}} \sum_{t=1}^{z_{jk}} mc_{-j}^k[1 + S_k - t]$. Putting the value of pay_j^p, we get the following expression: $\mathcal{U}_j^p = \sum_{k \in \mathcal{K}} \sum_{t=1}^{z_{jk}} mc_{-j}^k[1 + S_k - t] - \sum_{k \in \mathcal{K}} \overline{pp_k^j} z_{jk}$ which can be written as: $\mathcal{U}_j^p = \sum_{k \in \mathcal{K}} (\sum_{t=1}^{z_{jk}} mc_{-j}^k[1 + S_k - t] - \overline{pp_k^j} z_{jk})$. As earlier stated, as the providers are always truthful in terms of resource quantity, we can ignore the term z_{jk}. Then, the utility of a provider j becomes: $\mathcal{U}_j^p = \sum_{k \in \mathcal{K}} (\sum_{t=1}^{z_{jk}} mc_{-j}^k[1 + S_k - t] - \overline{pp_k^j})$. In order to prove the mechanism truthful, we consider different use cases and to prove that bidding truthfully is in best interest of any provider j.

(a) A cloud Provider wins i.e. non-zero quantity of VMs are allocated to the winning users in the auction. This case can be further categorized into two cases.
   - Among all possible allocations i.e. $z_{jk} = \{1, 2, \ldots, t, \ldots, qp_k^j - 1\}\ \forall k \in \mathcal{K}$ i.e. total t number of VMs are allocated to users where $t \leq qp_k^j - 1$ and

$mc^k_{-j}[S_k - t] < \overline{pp^j_k} < mc^k_{-j}[S_k - t + 1]$. If $pp^j_k = \overline{pp^j_k}$, then a total t number of VMs of type k are allocated among users. In this case, provider j will replace the cost of t units of highest cost VMs provided by other providers. If $pp^j_k < \overline{pp^j_k}$, then its chance of winning increases and more resources can be allocated if $mc^k_{-j}[S_k - t] < \overline{pp^j_k} < mc^k_{-j}[S_k - t + 1]$ is true. But this will decrease the overall utility of provider j. The reason is that as z_{jk} is increased, the value of $mc^k_{-j}[S_k - z_{jk} + 1]$ is decreased, as the values in mcf^j_k are in ascending order of marginal prices. Therefore, bidding less than its actual ask price lowers its utility. If $pp^j_k > \overline{pp^j_k}$, the winning chance of providers will decrease and there will always be less number of resources allocated as compared to the case when $pp^j_k = \overline{pp^j_k}$. If reported ask prices are higher than the highest marginal cost of VMs i.e. $pp^j_k > mc^k_{-j}[S_k]$, then the provider will lose and no VM will be allocated which bring its utility to zero. Therefore, in each case, a provider utility will either remain same or is decreased when it reports its ask prices falsely.

- $z_{jk} = qp^j_k \ \forall \ k \in \mathcal{K}$ i.e. a provider successfully allocates its full available capacity and it will replace the offers of other providers if its ask prices are lower i.e. $pp^j_k < mc^k_{-j}[S_k - z_{jk} + 1]$. If $pp^j_k = \overline{pp^j_k}$, then provider will allocate all of its available VMs if and only if $pp^j_k < mc^k_{-j}[S_k - z_{jk} + 1]$. Then, the provider will allocate all of its available capacity in order to increase its utility. The provider j will replace z_{jk} units of VMs of highest cost provided by other providers. If $pp^j_k < \overline{pp^j_k}$, there will be no change in the allocation as all the available capacity of provider j has been already allocated. Therefore, there will be no change in its utility. If $pp^j_k > \overline{pp^j_k}$, the winning chance of providers will be decreased and there will always be less number of resources allocates as compared to the case when $pp^j_k = \overline{pp^j_k}$. Therefore, in all three cases, the utility will remain same or is decreased if a provider bids falsely.

(b) Cloud provider loses when $\overline{pp^j_k} > mc^k_{-j}[S_k]$: If the true ask prices are higher than the highest marginal price of the resources, then the provider will not prefer to win in the auction which also can be achieved by bidding pp^j and qp^j truthfully. This is because the social welfare maximizing allocation would select the providers with the minimum cost and resource offers of provider j would not be allocated to any user i.e. $z_{jk} = 0 \ \forall k \in \mathcal{K}$.

1. If $pp^j_k > \overline{pp^j_k}$, then it will still be a loser in the auction because higher the valuation, lower the chance of winning in the auction. Therefore, its utility will remain zero. In this case also, false bidding brings no change in its utility.
2. If $pp^j_k < \overline{pp^j_k}$, then there are two further cases:
   - It may still be the loser when it does not qualify marginal cost of allocated quantities. Then, its utility will remain zero and there is no change in its utility by bidding falsely.
   - Suppose the provider lowers its valuations in such a way that $mc^k_{-j}[S_k - t] < \overline{pp^j_k} < mc^k_{-j}[S_k - t + 1]$ or $pp^j_k < mc^k_{-j}[S_k - z_{jk} + 1]$. In this case its payment will be equal to $mc^k_{-j}[S_k - t + 1]$ and $mc^k_{-j}[S_k - t + 1] < \overline{pp^j_k}$. Therefore, utility of the provider will be negative in this case. This case also proves a decrease in its utility by bidding falsely.

All the above cases prove that if the provider bids different than its actual valuations (whether greater than or less than), then its utility either remains the same or is decreased. If $pp^j_k < mc^k_{-j}[S_k - z_{jk} + 1]$, provider j will allocate all of its available VMs in order to maximize the $\mathcal{U}^p_j$, which can also be achieved by truthfully reporting pp^j_k and z_{jk}. If $mc^k_{-j}[S_k - t] < \overline{pp^j_k} < mc^k_{-j}[S_k - t + 1]$ where $t \in \{1, 2, \ldots, t, \ldots, qp^j_k - 1\}$, the provider will prefer to allocate t units of VMs in order to maximize its utility which can also be achieved by reporting pp^j_k and z_{jk} truthfully. If provider's reported cost are higher than the highest marginal cost of allocated quantity i.e. $\overline{pp^j_k} > mc^k_{-j}[S_k]$, then it will not prefer to offer any VM because it will results in a negative utility. Therefore, it will remain truthful to obtain non-negative utility. This proves that bidding truthfully becomes the providers' dominant strategy. This proves that the proposed mechanism is incentive compatible for all cloud providers. □

Theorem 5. *TCMDAC is Individual rational for all cloud providers.*

Proof. In TCMDAC, if a cloud provider wins, it always gets paid an amount which is always greater than or equal to its actual ask price/cost. If it doesn't win, it will get nothing. This always results in a non-negative utility for the provider i.e. $\mathcal{U}^p_j \geq 0 \ \forall j \in \mathcal{M}$. Therefore, the proposed mechanism is individual rational for all providers. □

Theorem 6. TCMDAC *is WBB (weak budget-balance).*

Proof. Budget-balance property ensures that total payment done by the users should be equal or greater than the total payment received by the providers. i.e. $\sum_{i \in \mathcal{N}} pay^i - \sum_{j \in \mathcal{M}} pay^j \geq 0$. To prove the budget-balance, lets first calculate the lower bound on the payments done by all winning users and then calculate the upper bound on the payments or rewards paid to all winning providers. If the difference is positive, then the mechanism is weak budget-balance. By the definition of $criticalValue^u_i$, for user i, if $pu_i > criticalValue^u_i$, $x'_i = 1$ in the optimal solution of (x', z') to $\bar{\Pi}(\mathcal{N}, \mathcal{M}, \mathcal{V})$. As long as the $pu_i > criticalValue^u_i$, the solution (x', z') remains same and also the surviving user set $\mathcal{N}^s$. Also, according to Lemma 1, $pu_i > \sum_{k \in \mathcal{K}} qu^i_k * mc^k[\sum_{i \in \mathcal{N}} qu^i_k x'_i + qu^{\mathcal{V}}_k] \geq \sum_{k \in \mathcal{K}} qu^i_k * mc^k[\sum_{i \in \mathcal{N}^s} qu^i_k + qu^{\mathcal{V}}_k]$. Therefore, as long as pu_i approaches $criticalValue^u_i$ the term $\sum_{k \in \mathcal{K}} qu^i_k * mc^k[\sum_{i \in \mathcal{N}^s} qu^i_k + qu^{\mathcal{V}}_k]$ remains unchanged because (x', z') remains same. Therefore, in the limit case, $criticalValue^u_i \geq \sum_{k \in \mathcal{K}} qu^i_k * mc^k[\sum_{i \in \mathcal{N}^s} qu^i_k + qu^{\mathcal{V}}_k]$. Also $\sum_{i \in \mathcal{N}^s} qu^i_k = \sum_{j \in \mathcal{M}} z''_{jk}$ where $z''_{jk} \leq qp^j_k$. In the final allocation (x'', z''), total $\sum_{i \in \mathcal{N}^s} qu^i_k$ units of VMs of type $k \in \mathcal{K}$ are allocated to the winning users. The payments from all the users is no less than the total payments when $\sum_{i \in \mathcal{N}^s} qu^i_k$ units of VM of type $k \in \mathcal{K}$ are traded at per unit price $mc^k[\sum_{i \in \mathcal{N}^s} qu^i_k + qu^{\mathcal{V}}_k]$ for all $k \in \mathcal{K}$. After setting the lower bound for user side, we consider the provider side and find out an upper bound for payments received by all providers. According to Lemma 1, the maximum payment of per unit of traded VM of type k is $mc^k_{-j}[S_k] = mc^k_{-j}[\sum_{i \in \mathcal{N}^s} qu^i_k]$. As mc^k and mc^k_{-j} are monotonic increasing function and $qu^{\mathcal{V}}_k > 0 \ \forall \ k \in \mathcal{K}$, therefore $mc^k_{-j}[\sum_{i \in \mathcal{N}^s} qu^i_k] \leq mc^k[\sum_{i \in \mathcal{N}^s} qu^i_k + qu^{\mathcal{V}}_k] = mc^k[\sum_{j \in \mathcal{M}} z'_{jk}]$. Accordingly, the total payments received by all providers i.e. $\sum_{j \in \mathcal{M}} pay^j$is no more than the total payment when $\sum_{i \in \mathcal{N}^s} qu^i_k$ units of VM of type $k \in \mathcal{K}$ are traded at per unit price $mc^k[\sum_{i \in \mathcal{N}^s} qu^i_k + qu^{\mathcal{V}}_k]$ for all $k \in \mathcal{K}$. Therefore, total payment paid by all users is always greater than or equal to the total payments paid to all providers i.e. the difference between the revenue from the users and the payments to the providers is always non-negative. Thus, the proposed TCMDAC mechanism is weak budget-balanced. □

Theorem 7. *TCMDAC is computationally tractable.*

Proof. Here, in this section the worst-case complexity of TCMDAC is discussed. TCMDAC model mainly contains two phases i.e. allocation phases and pricing phase. In allocation phase, VPU is considered in problem $\bar{\Pi}(\mathcal{N}, \mathcal{M}, \mathcal{V})$. VPU's resource quantity determination phase would take O(MK) as there are total K types of VMs and there are total M providers and for each type of VM, maximum quantity is chosen by screening all providers' offered quantity. Solving problem $\bar{\Pi}(\mathcal{N}, \mathcal{M}, \mathcal{V})$ takes polynomial time O(p) as it can be solved using any Linear Programming (LP) based methods. Therefore, the total computational time complexity of the allocation phases is $O(MK) + O(p)$. The payment phase further contains two different pricing mechanisms. These schemes use the marginal values of resources provided by various providers. Marginal cost ($mc^k_{-j}[q]$) construction takes $O(KM\log(M))$ as all providers are sorted according to their ask prices and this repeats for each type of VM. User's payment is derived using critical value pricing by doing binary search over a fix range whose upper bound is a user valuation. Binary search complexity is $O(\log(\frac{1}{\epsilon}))$, if the given accuracy is ϵ. For each user, $\bar{\Pi}(\mathcal{N}, \mathcal{M}, \mathcal{V})$ is solved with complexity O(p). Therefore, total complexity of user's pricing mechanism will be $(N\log(\frac{1}{\varepsilon})) \times O(p)$. For providers' payment calculation, marginal values are directly used in the payment process. One extra thing is to calculate the total allocated quantity i.e. S_k for each type of VM which have $O(M+Z)$ where Z is the upper bound of z_{jk}. Therefore, the total computational time complexity of pricing of providers is $O(M+Z)$. In conclusion, upper bound of overall complexity of TCMDAC is $O(KM\log(M)) + (N\log(\frac{1}{\varepsilon}))O(p)$ which is polynomial. Therefore, TCMDAC is computationally tractable. □

Theorem 8. *TCMDAC is asymptotically efficient as the providers' total resources compared with resource demand becomes more and more sufficient, given bounded cost distributions of workload for all VM types.*

Proof. We have to prove that the allocation (x, w) converges to optimal solution(x^*, w^*) of social welfare maximization problem $\bar{\Pi}(\mathcal{N}, \mathcal{M})$ on the condition that the total offered resources becomes more and more efficient. To prove this, we first fix the number of users and their demand. After that we increase the number of resources provided by the cloud providers and check the total social welfare generated afterwards. We assume that ask price is randomly generated between the interval $[\underline{pp^j_k}, \overline{pp^j_k}]$. According to proposed allocation method as $pu_i < \sum_{k\in\mathcal{K}} qu^i_k \underline{pp^j_k}$ then in that case, $x'_i = x''_i = x^*_i = 0$ i.e. in both methods, user i does not win in auction. If we increase the supply, there may be a cases that more providers with cheap ask price will offer their resources. In that case, the inequality may be true: $pu_i > qu^i_k * mc^k[\sum_{i\in\mathcal{N}} qu^i_k x'_i + qu^{\mathcal{V}}_k] > \sum_{k\in\mathcal{K}} qu^i_k \underline{pp^j_k}$. By proposed mechanism, user i will win in TCMDAC. As $pu_i > qu^i_k * mc^k[\sum_{i\in\mathcal{N}} qu^i_k x'_i + qu^{\mathcal{V}}_k]$, user i also wins in optimal allocation. Further, when more and more resource capacity is available, more providers with their lower pp^j_k values offer their resources which helps some users to win in the auction. This results in the increment in the total social welfare. When offered resource is enough, (x, z) and (x^*, z^*) are equal. □

Theorem 9. *TCMDAC mechanism is incentive compatible, individual rational, weakly budget balanced and asymptotic efficient if* $qu^{\mathcal{V}}_k = max_{j\in\mathcal{M}}\{qp^j_k\}\ \forall\ k \in \mathcal{K}$.

Proof. From Theorem 1–8, it can be shown that TCMDAC is incentive compatible, individual rational, weakly budget balanced, computationally feasible and asymptotic efficient.

The above properties also hold true even when $qu^{\mathcal{V}}_k > max_{j\in\mathcal{M}}\{qp^j_k\} - 1\ \forall\ k \in \mathcal{K}$. The VPU's resource quantity that has been considered in TCMDAC is $qu^{\mathcal{V}}_k = max_{j\in\mathcal{M}}\{qp^j_k\} - 1\ \forall\ k \in \mathcal{K}$. We consider the maximum quantity of offered resources from all providers because a provider with the largest resource offers has the largest power in manipulating the price. Accordingly, a VPU $\mathcal{V}$ with $qu^{\mathcal{V}}_k > max_{j\in\mathcal{M}}\{qp^j_k\} - 1\ \forall\ k \in \mathcal{K}$ may also be used for ensuring truthfulness and budget-balance. But large quantity can cause larger loss of efficiency. Therefore, we have considered the minimum quantity values of VPU among all possible quantity values.□

5. Performance evaluation

Evaluating the performance of the proposed model (user satisfaction, cost benefits, revenue etc.) on real cloud environments such as Amazon EC2 (Amazon, 2016), Google App Engine, Microsoft Azure (Microsoft, 2016) etc. is tedious, challenging and time consuming process (Calheiros et al., 2011). The reason is that cloud exhibits varying supply and demand patterns and contains heterogeneous resources, whereas cloud users have heterogeneous and competing resource requirements. Therefore, the performance evaluation is constrained by infrastructure's rigidity. Further, it is very complicated and time consuming process to re-configure the benchmarking parameters across cloud infrastructure over multiple test runs (Calheiros et al., 2011). Another approach to evaluate the performance is to use CloudSim (Calheiros et al., 2011) which is a well-known java based simulator though it doesn't support the auction implementation. CloudSim is extended by CloudAuction (Samimi et al., 2016) that creates environment for combinatorial double auction. But this supports greedy based resource allocation and first price only. Proposed model can't be evaluated using CloudAuction (Samimi et al., 2016) and CloudSim (Calheiros et al., 2011) because it is based on LP-based padded optimization method and truthful pricing using critical and marginal values.

Because of unavailability of dedicated simulator and real data, TCMDAC has been simulated in MATLAB. We considered different cloud market environment and analyzed and compared the performance of TCMDAC with other models. The resource demand was kept high and resource supply low as it best suits the current cloud market scenario with less number of cloud providers in comparison to number of cloud users. Each experiment was run for 20 iterations and the average is taken for all performance measures. TCMDAC is compared with four models: CDARA (Samimi et al., 2016), DS-VRAP (Chichin et al., 2015a), CDAGC and OPTIMAL mechanism.

Details of models used for comparison

CDARA: Samimi et al. (2016) proposed a double auction based resource allocation model named CDARA for cloud computing environment. Samimi et al. (2016) applied greedy heuristics for allocating the resources in approximated manner. CDARA model adopts the average pricing mechanisms earlier proposed in grid settings (Li et al., 2009). Here, the noticeable point during allocation is that a particular type of VM requested by a user is matched with all types of VMs available with cloud providers by comparing respective resource attribute values. Our allocation method is different from CDARA. In our proposed model TCMDAC, we abstract the resource attribute values and derive a weight value for each type of VM using one or more resource attribute values as described in the problem model. Therefore, we have modified the CDARA algorithm by adding weights to VMs and abstracting the resource attributes values. During allocation, we match the supply and demand for each type of VM separately. The main modification is that during allocation, a particular type of VM requested by user is matched with the same type of VM offered by a cloud provider. As CDARA considers combinatorial bid on provider side also, we have generated a single combinatorial bid using some discount values from the input bids. For example, if a provider provides 3 VMs of type 1 and 2 VMs of type 2 in offering and their per-unit, ask price is 1$ for type 1 VM and 2$ for type 2 VM. As the bundle price is always less than the total price of items, suppose discount rate is 20% of the total cost. Then in that case, combinatorial bid value will be $(3 \times 1\$ + 2 \times 2\$) \times 0.8 = 5.6\$$.

DS-VRAP (Chichin et al., 2015a): Another model that we have considered for comparison is DS-VRAP (Double sided Virtual Resource Allocation Problem) proposed in Chichin et al. (2015a). In DS-VRAP, a combinatorial greedy allocation scheme is proposed. Here, a notion of *candidate* is introduced and there can be a maximum $M \times N$ candidates where M and N are the number of providers and users. These candidates are sorted using $candidate - surplus\ density\ value$ which is defined as the surplus induced by the candidate per unit of traded resource. For cloud users, truthful pricing mechanisms using critical payment is proposed whereas two payment schemes i.e. proportional value based and user based pricing schemes is proposed for providers. The proportional payments schemes were earlier proposed in Stößer et al. (2010). In user based pricing scheme for provider, users' payment is directly paid to providers based on the offered resource quantity. In proportional value based pricing mechanism, surplus generated from the market is distributed among all winning providers proportional to their contributions. The double auction mechanism proposed in DS-VRAP satisfies the truthfulness for users only. Also, the model generates approximated result. The proposed model considers certain constraints such as request indivisibility, resource bundling and demand aggregation. In the model, a cloud user can get its resources from a single provider only whereas a single provider can allocate its resources to multiple users. In TCMDAC, we relax the constraint of request indivisibility i.e. we assume that a user can acquire its resources from multiple providers.

CDAGC: In order to do detailed analysis and comparison, a model of combinatorial double auction is designed inspired from CDARA (Samimi et al., 2016), DS-VRAP (Chichin et al., 2015a) and (Zaman and Grosu, 2013) named as CDAGC (**C**ombinatorial **D**ouble **A**uction with **G**reedy Allocation and **C**ritical Payments in cloud Computing). CDAGC consists of two parts: allocation function and payment scheme. Allocation is similar as in CDARA except one modification. CDAGC considers the constraint of request indivisibility i.e. if a cloud user wins, it can obtain its resources from a single provider only. CGAGC's Payment mechanism for seller is user-based pricing scheme(S-BPS) which was used previously in Chichin et al. (2015a). For winning users, critical payments were calculated using the method proposed in Zaman and Grosu (2013) which ensures that CDAGC is truthful for all cloud users.

OPTIMAL: We compared the results of TCMDAC with OPTIMAL mechanism to find the gap between the TCMDAC solution and Optimal solution. In OPTIMAL mechanism, social welfare maximization problem i.e. $\Pi(\mathcal{N}, \mathcal{M})$ is solved in an optimal manner (Xia et al., 2005). For payments, first pricing is used for both user and provider (Klemperer, 2004). In first price double auction, if a user wins, it pays its quoted bid price whereas a provider gets the amount equal to its quoted ask price multiplied by the offered resource quantity. Although this mechanism optimally allocates the resources among users and providers, but the mechanism is not truthful for any participant. Table 2 compares all the models in terms of auction properties and their assumptions.

5.1. Simulation settings and data generation

In current cloud market, each provider has some service limitations, e.g., Microsoft Azure has default limitation of maximum 20

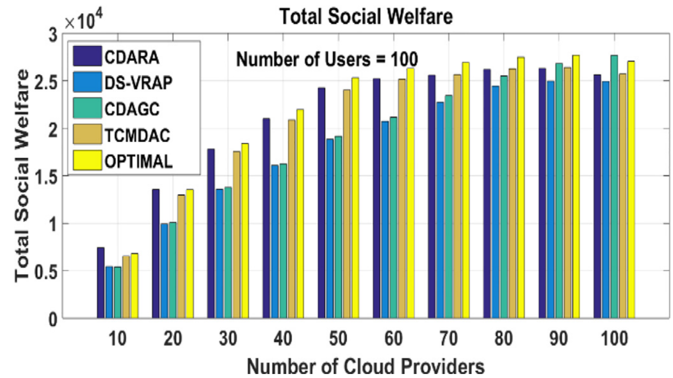

Fig. 1. Total social welfare with increasing number of providers.

VMs for each user (Microsoft, 2016). Therefore, we fixed the maximum number of VMs provided by a provider to 10 and the resource quantity offered is uniformly generated in the range [1,10]. In order to generate feasible allocation for CDAGC and DS-VRAP, the resource quantity required by users is uniformly generated in the range [1,5]. A cloud market is considered with various providers and 4 different types of VMs. For providers' bid generation, we analyzed the VM pricing of real cloud providers. For this, the data is collected from CLOUDORADO (cloudorado, 2016) which is an online platform for comparing all cloud providers in terms of their resource configuration and pricing. We observed the pricing patterns of various types of VM from all cloud service providers and generated the prices of VM accordingly. To give the importance to each VM, the weights for differentiating various types of VMs are considered as $w_1 \le w_2 \le \cdots \le w_K$ i.e. weights are upward scaled. The value of w_1 is randomly taken between $[1, 2]$, w_2 is randomly taken between $[w_1 \times 1.5, w_1 \times 1.8]$, w_3 is randomly taken between $[w_2 \times 1.5, w_2 \times 1.8]$ and so on. On the provider side, the weighted VM concept is implemented in a similar manner. Since $pp_1^j < pp_1^j < \cdots pp_K^j$, where pp_k^j be the price of VM of type k where $k \in \mathcal{K}$, the values of pp_1^j is randomly generated between $[1, 2]$, pp_2^j between $[pp_1^j \times 1.5, pp_1^j \times 1.8]$, pp_3^j between $[pp_2^j \times 1.5, pp_2^j \times 1.8]$ and so on. In double auction it is necessary that a user's valuation should be greater than the equivalent resource bundle cost of offering provider.

We assume that most of the allocations satisfy the above condition. Therefore, larger users' bid values have been taken in experiments and generated using uniform distribution in the range [100, 500]. Table 3 presents the parameter settings for the simulation.

5.2. Results

For the exhaustive evaluation, the whole study is done in different market settings. As the double auction based mechanism depends upon the demand and supply, we consider different values for it in the experiments. First, number of users is fixed and number of providers is varied. After that, the number of providers is fixed and the number of users is varied. Then, both the number of users and providers are varied in equal proportion. Lastly, various degree of competition on both sides of the market is enabled to analyze the behavior of TCMDAC model by applying different demand and supply.

In the first experiment, the number of users is fixed to 100 and the number of providers is varied from 10 to 100. Various performance metrics have been used to compare the performance of TCMDAC and other models. Figs. 1 and 2 show the social welfare and users' satisfaction of all five models with increasing number of cloud providers. Here, users' satisfaction is used as a performance metric which is calculated as the ratio of total valuation of winning

Table 2
Comparison of TCMDAC with other models in terms of auction properties and system settings.

Property/model	CDARA	DS-VRAP	CDAGC	OPTIMAL	TCMDAC
Allocation	Greedy	Greedy	Greedy	ILP based optimal allocation	LP and padding based allocation
Users' payment	Average pricing	Critical payment	Critical payment	First pricing	***Critical value based payment***
Providers' payment	Average pricing	Proportional value pricing	Buyer based pricing	First pricing	***Marginal cost based payment***
Request indivisibility	No	Yes	Yes	No	No
Truthfulness for user	No	Yes	Yes	No	***Yes***
Truthfulness for provider	No	No	No	No	***Yes***
Individual rational	No	Only for cloud users	Yes	Yes	***Yes***
Budget-balance	Strong	Weak	Weak	Weak	***Weak***
Asymptotic efficient	No	No	No	Yes	***Yes***
Combinatorial bidding – provider	Yes	No	No	No	No

Table 3
Simulation parameters.

Parameter types	Description
Types of VMs	4 (VM1, VM2,VM3,VM4)
User's requested quantity	VM1[1,5], VM2[1,5], VM3[1,5], VM4[1,5]
User's bid valuation	[100,500]
Weight vector	VM1[1,2],VM2[w1 × 1.5,w1 × 1.8], VM3[w2 × 1.5,w2 × 1.8], VM4[w3 × 1.5,w3 × 1.8]
Provider's ask prices	VM1[1,2],VM2[p1 × 1.5,p1 × 1.8], VM3[p2 × 1.5,p2 × 1.8], VM4[p3 × 1.5,p3 × 1.8]
Provider's offered quantity	VM1[0,10], VM2[0,10], VM3[0,10], VM4[0,10]
Number of iteration	20

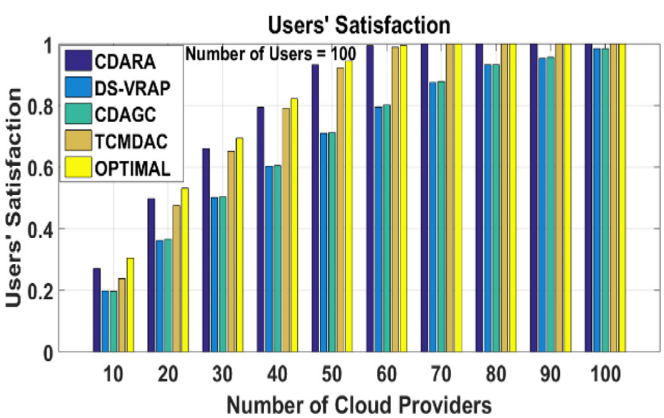

Fig. 2. Users' satisfaction with increasing number of providers.

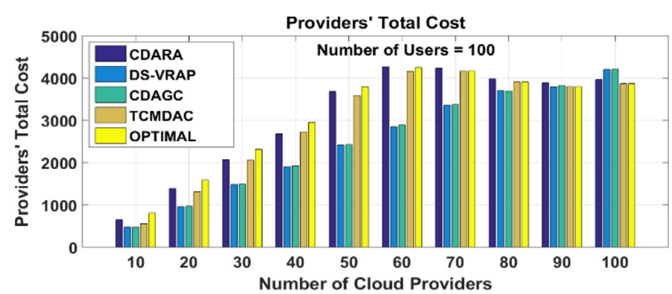

Fig. 3. Providers' total cost with increasing number of providers.

users to the total valuation of all users i.e. $\sum_{i\in\mathcal{N}} pu_i x_i / \sum_{i\in\mathcal{N}} pu_i$. As number of cloud provider increases, more resources would be available in the cloud market and will serve the users more effectively leading to more winning users and higher social welfare in all the models.

From Figs. 1 and 2, it can be observed that the social welfare and user's satisfaction in CDARA and TCMDAC are nearly equal and outperforms DS-VRAP and CDAGC models. The reason is that in DS-VRAP and CDAGC, a user can acquire its resources from a single provider only. Therefore, some of the possible allocations are rejected in these two models. As the OPTIMAL mechanism solves the social welfare maximization problem in an optimal manner, it generates highest social welfare among all mechanisms. Another reason for less social welfare of TCMDAC as compared to OPTIMAL method is that some of the efficient allocations/trades are sacrificed in order to achieve incentive compatibility for all participants. The social welfare and users' satisfaction of CDARA, TCMDAC and OPTIMAL does not change when providers are increased from 60–70 to 100. It is because when the number of providers reaches 60–70, demand of resources equals the supply resulting in no further allocations. In CDAGC and DS-VRAP, resources are not fully utilized due to the constraint of request indivisibility. Therefore, when the number of providers in the market increases, more allocations happen among users and providers, which further increases the total social welfare and the users' satisfaction.

Fig. 3 shows the providers' total cost of allocated resources of five models which first is increased and then is decreased for CDARA, TCMDAC and OPTIMAL. The reason for such a behavior is that during allocation, CDARA, TCMDAC and OPTIMAL select the provider with the minimum ask price. When demand is less and supply of resources is more, resources with lower cost are allocated to the users. When the number of cloud providers is increased, resource availability further is increased and more providers with lower costs are available to offer resources which results in an overall lower total reported costs/ask prices. This doesn't happen in the case of DS-VRAP and CGAGC. The reason is less allocation in CDAGC and DS-VRAP due to request indivisibility constraint even after oversupply of the resources. Therefore, when the number of providers are further increased, more allocations happen and more providers are able to win the auction increasing the providers' total cost.

Users' total payment and Providers' revenue with varying number of providers are examined in Figs. 4 and 5. It can be seen that the OPTIMAL method results in highest payment because of the first price schemes. TCMDAC, CDAGC and DS-VRAP payment schemes are truthful and users will always pay less than their actual valuation. When number of providers are less, i.e. requested quantities is more than the supply, TCMDAC generates higher payment. But when the number of providers is increased, more users with lower valuations wins in the auction. This results in lowering their critical payments thus performing better when the sup-

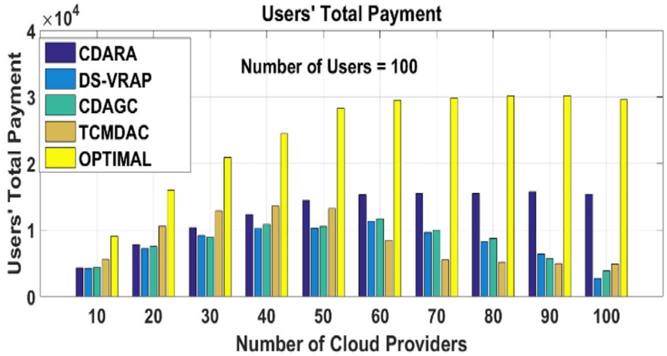

Fig. 4. Users' total payment with increasing number of providers.

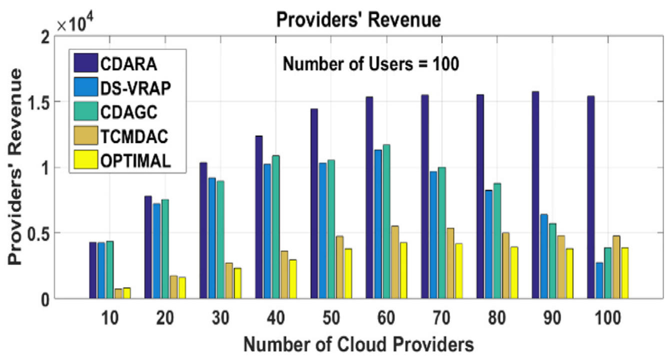

Fig. 5. Providers' total revenue with increasing number of providers.

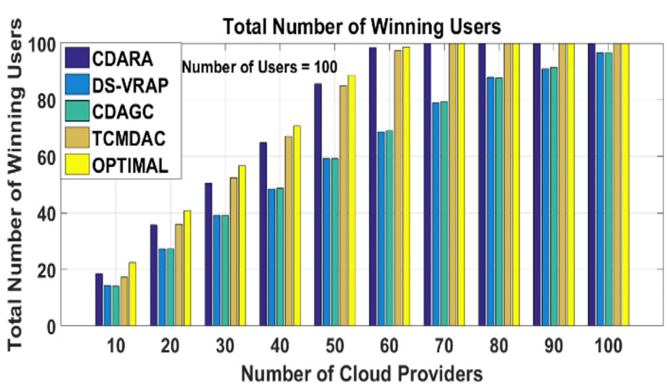

Fig. 6. Total number of winning user with increasing number of providers.

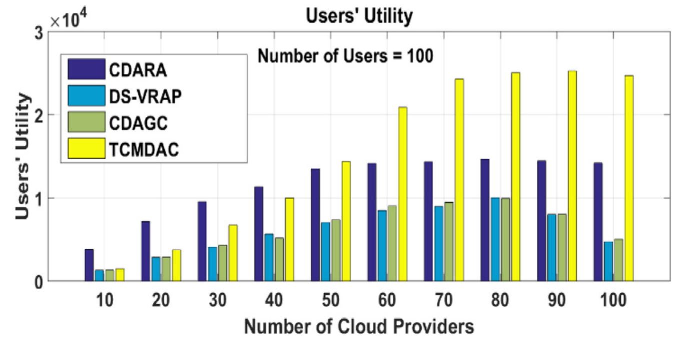

Fig. 7. Users' total utility with increasing number of providers.

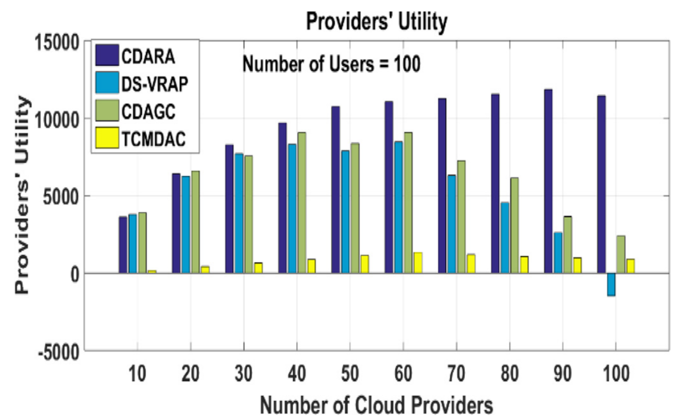

Fig. 8. Providers' total utility with increasing number of providers.

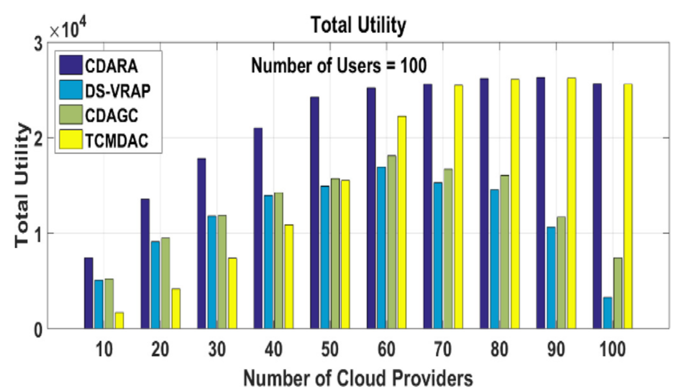

Fig. 9. Total utility in market with increasing number of providers.

ply is efficient. When demand equals supply, total payment does not change further with increasing number of providers. Further, it can be observed that the user's payment of CDARA is not decreased due to average pricing. In case of provider revenue, it first increased then decreased when offered resources becomes more than required quantity. This is due to the fact that more providers with lower cost/ask prices will be able to offer the resources which lowers the total revenue of providers. Same as above, OPTIMAL generate least revenue because each provider will get paid the amount equal to its actual cost/ask price multiplied by offered quantity.

We examine the relationship between the number of winning users and number of providers for all five models with varying number of providers as shown in Fig. 6. As the number of providers is increased, more resources available for allocation which results in a higher number of winning users for all the models. When providers reach 60–65, requested quantities of resources equals the total offered resource quantity. At this situations, all users in TCMDAC, CDARA and OPTIMAL successfully acquires the resources and win in the auction and will remain same when the number of providers further increased. In case of CDAGC and DS-VRAP, less users will win due to the request indivisibility constraint and number of winning users keeps on increasing with the number of providers.

Fig. 7, Fig. 8 and Fig. 9 presents the relationship between the utility of the participant and the number of providers in the market. When number of providers is low, TCMDAC results in lesser users' total utility as compared to CDARA model because some allocations are rejected in order to achieve truthfulness. When the number of providers are increased, almost all users win the auction. TCMDAC method has highest utility as it has lesser payment for all winning users as compared to CDARA and other models. Further, the utility does not change when the number of providers are increased from 70 to 100. In case of providers' utility, three pricing mechanisms i.e. average pricing, proportional value pricing and buyer based pricing depend upon the buyer's valuations

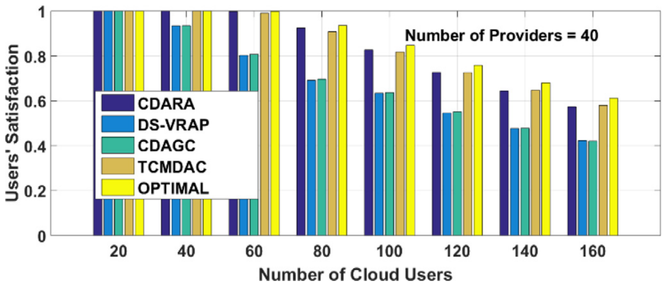

Fig. 10. Users' satisfaction with increasing number of users.

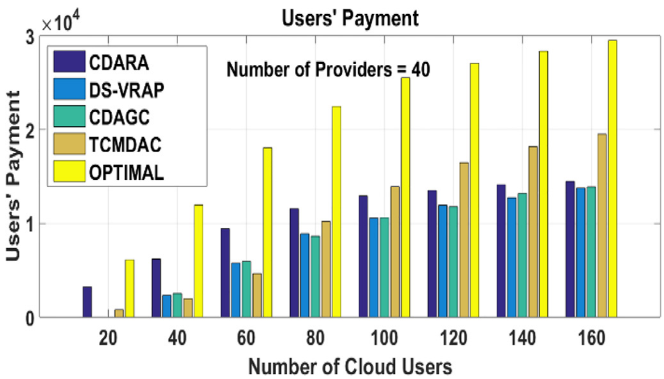

Fig. 11. Users' total payment with increasing number of users.

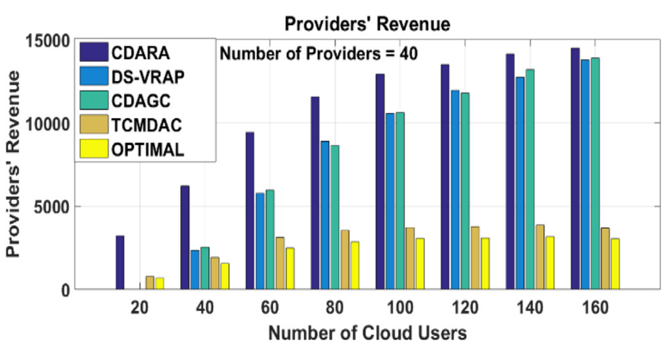

Fig. 12. Providers' total revenue with increasing number of users.

thus generate higher revenue and utility for winning providers as compared TCMDAC. TCMDAC generates lowest revenue among all models and also the lowest Providers' utility. In terms of total utility, when number of users is more and number of providers is less, TCMDAC performs poor as compared to other models when there are less available resources or less providers in the market. When the number of providers are increased, TCMDAC's total utility is also increased and outperforms other models. Fig. 9 also verifies the theoretical result that TCMDAC performs better when there is sufficient supply of resources.

After considering varying number of providers, the behavior of TCMDAC and other models are examined with different number of users. The performance metrics considered are users' satisfaction, users' payment and providers' revenue in this case. The number of providers is fixed to 40 and the users are varied from 20 to 160. Fig. 10 examines the relationship between the users' satisfaction and the number of users. Here, notable is that users' satisfaction is decreased as the number of cloud users is increased. The reason for this being that more cloud users make the competition more intense and more competitive users i.e. users with low valuation also win in the final allocation, hence lowering the total valuation. The above result is true for all five models. TCMDAC method has

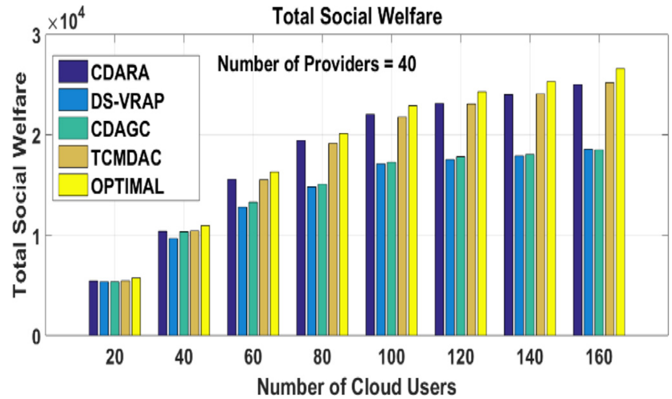

Fig. 13. Total social welfare with increasing number of users.

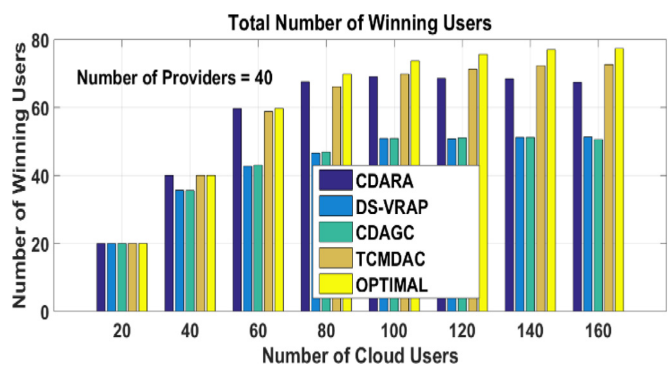

Fig. 14. Total winning users with increasing number of users.

almost same user's satisfaction as OPTIMAL method but outperforms CDAGC and DS-VRAP models.

Figs. 11 and 12 show the change in users' payment and providers revenue according to change in the number of users. As winning users in CDAGC, DS-VRAP and TCMDAC pay the critical payment, the total payment would be less as compared to OPTIMAL method in which all winning users pay their actual valuations. Average pricing generates more payments as compared to CDAGC and DS-VRAP due to more number of winning users and more allocations. As number of users in the cloud market are increased, more resources will be allocated to the requesting users which results in higher user payment. In case of providers' revenue, OPTIMAL offers lowest revenue because of first pricing. Here also, TCMDAC generates lower payments for winning providers for their offered resources. It is to be noted that TCMDAC generates truthful payments for providers in contrast to all other models.

Fig. 13 presents the total social welfare of all five models with varying number of users. It shows that TCMDAC performs well as compared to other three models in terms of social welfare. The social welfare generated by TCMDAC method is nearly equal to CDARA and OPTIMAL method. As the number of cloud users are increased, more and more resources are allocated to this increasing demand which eventually increases the total social welfare. As TCMDAC and OPTIMAL method allocate the resource in an optimal manner, both of these methods would result in allocation with higher number of winning cloud users as compared to CDAGC and DS-VRAP as shown in Fig. 14. In CDAGC and DS-VRAP, the number of winning users will be almost the same but lesser as compared to TCMDAC, CDARA and OPTIMAL. Here also, OPTIMAL results in highest number of winning users as it optimally allocates the offered resource to the requesting users.

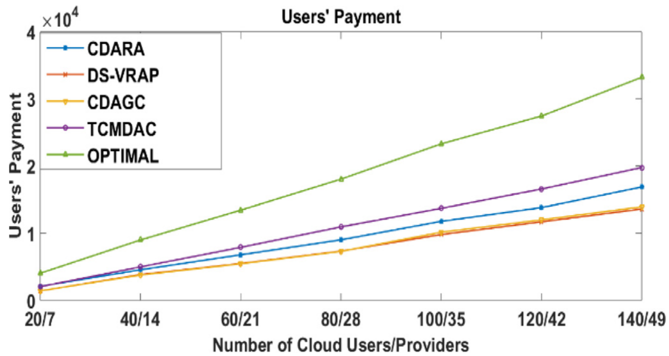

Fig. 15. Users' payment with increasing market size.

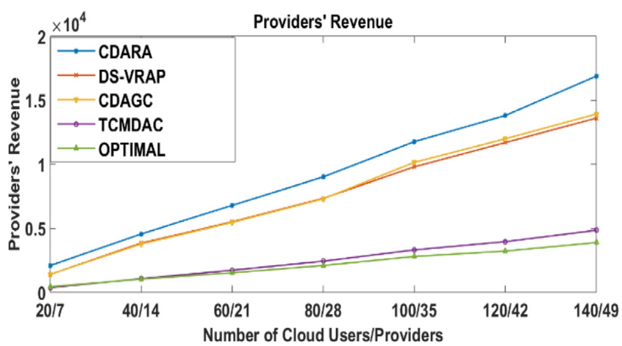

Fig. 16. Providers' revenue with increasing market size.

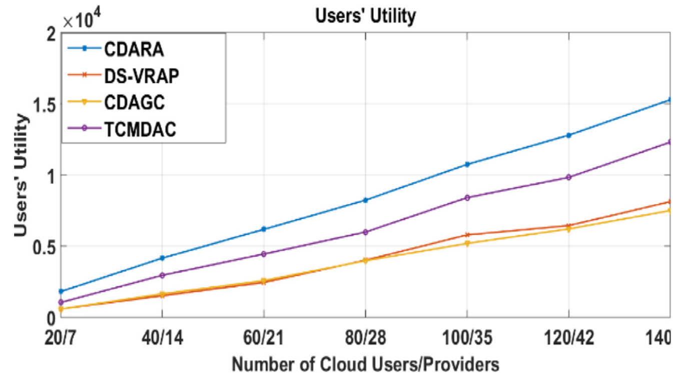

Fig. 17. Users' utility with increasing market size.

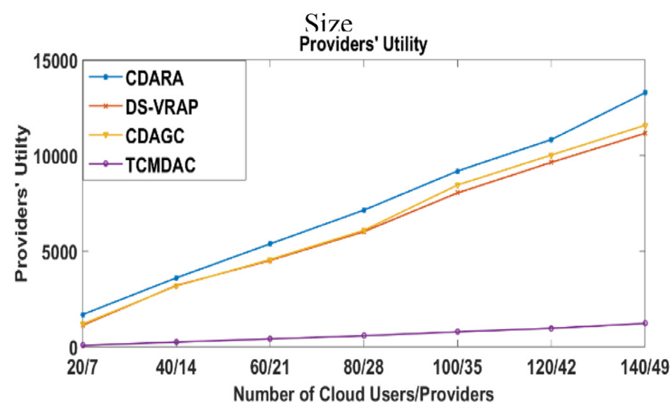

Fig. 18. Providers' utility with increasing market size.

We increase the market size by increasing number of users and providers in an equal proportion and observe the behavior of TCMDAC and other four models with increasing market size.

Figs. 15 and 16 show the users' total payment and providers' revenue with increasing market size. From Fig. 15, it can be observed that the OPTIMAL mechanism performs the worst by generating highest users' payment among all five models. In OPTIMAL method, first price mechanism is used for winning users i.e. a winning user pays the amount equal to its actual valuation. Therefore, it results in highest total payment among all the models. As TCMDAC, DS-VRAP and CDAGC methods have critical payment based pricing mechanisms (critical payments are always less or equal to its actual valuations), the total users' payment will always less than the OPTIMAL method. CDARA results in higher payment value due to average pricing. In case of provider revenue, the average pricing, proportional value pricing and buyer based pricing mechanisms proposed in CDARA, DS-VRAP and CGAGC respectively produces higher revenue for winning providers as compared to TCMDAC and OPTIMAL but these models are neither truthful nor individual rational. Among all five models, TCMDAC is the only model which is truthful for cloud providers. In first price auction as adopted in OPTIMAL mechanism, a provider will get paid the amount equal to their actual ask prices of resources. The optimal method with first price auction will always produce least revenue for providers. As TCMDAC is individual rational and truthfulness, a provider will get paid higher than its actual cost/ask price. Fig. 2 clearly shows that TCMDAC produces higher revenue for providers as compared to OPTIMAL.

Figs. 17 and 18 present the utility of cloud users and cloud providers with varying number of users and providers. As mentioned earlier, incentive compatibility and efficiency cannot be achieved together in a double auction mechanism. If a mechanism is truthful, it loses some efficiency. Similarly, if a mechanism is efficient, it can't ensure truthfulness. As TCMDAC mechanism ensures the truthfulness for both the cloud users and providers, it loses some efficiency which results in lower utility as compared to other mechanisms. Further it can be seen from Fig. 17 that user's utility of TCMDAC is more as compared to DS-VRAP and CDAGC because DS-VRAP and CDAGC result in lesser allocations as compared to TCMDAC. In case of providers' utility, TCMDAC exhibits lowest utility because of truthful pricing. CDAGC, CDARA and DS-VRAP models are not incentive compatible for providers and the pricing mechanisms proposed for these have higher payment values as already shown in Fig. 16 which results in higher utility for the of providers. Although providers' total utility in TCMDAC is lowest among all models, yet it is the only model which is truthful for both user and the provider.

We examine the performance of TCMDAC model by varying the supply and demand of resources available in the market. For this, we take $N=50$ and $M=20$. Different degree of competition on both sides of market is considered to evaluate its effect on users' satisfaction. Three patterns for demand i.e. half demand, normal demand and double demand are considered. The same pattern is adopted for supply of resources by considering half supply, normal supply and double supply of resources. The normal demand/supply is as default bids/offers, whereas half demand/supply is calculated by cutting their total demand/supply to half. In a similar manner, double demand/supply is twice of default bids/offers. Fig. 19 shows that user's satisfaction is decreased when the users' requests are increased. This indicates that users are harder to win when competition on their side increases. Another notable point is that users' satisfaction is increased when the resource availability is increased. This clearly shows that a user's chance of winning is increased when the competition on provider side is increased. The above results demonstrate that TCMDAC competition on one side of market is favorable for players on the other side of market whereas unfa-

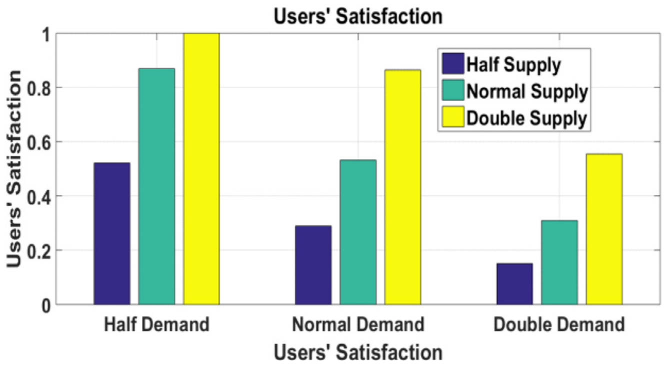

Fig. 19. users' satisfaction with various demand/supply ($N = 50$, $M = 20$).

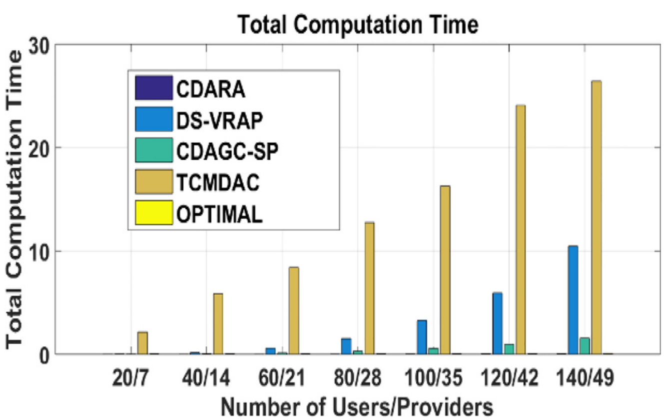

Fig. 20. Total computation time (allocation + payment) with increasing market size (in seconds).

vorable for players on its own side of market which is natural and rational for two-sided market.

The computation time (Allocation + Payment) observed for all five models is shown in Fig. 20. It can be seen that, CDARA model takes very less time because of its greedy allocation scheme and average pricing mechanism. DS-VRAP and CDAGC take considerably more time due to critical payment calculation. This is because for each winning user, first the user is removed from user's participating list and re-allocation is done without considering that user. Therefore, each winning user's payment calculation requires re-allocation thus increasing the total computation time of mechanism. TCMDAC would take highest computation time among all models due to its LP-based allocation mechanism and truthful payment calculations for all participants. As TCMDAC ensures truthfulness at both sides, the payment calculation and inclusion of VPU will take more time as compared to other methods. Here, cloud users' payment is determined using critical values which are calculated by binary search over a determined range of values. The whole payments' calculation process contains several re-allocations which increases the total computation time of TCMDAC. For each winning provider, payments are calculated using updated marginal values. OPTIMAL mechanism considers one-shot allocation optimally and first pricing calculation both of which takes less time in comparison to others.

From the aforementioned discussions and analysis, we can now draw the conclusion that TCMDAC is an efficient and truthful mechanism to allocate the cloud resources under the settings discussed in this paper. To make any mechanism truthful, we have to give some incentives to participants to reveal the truth which in turns reduces the utility (Narahari, 2014). Therefore, TCMDAC lacks in some metrics such as total utility and efficiency in some settings, yet it is acceptable mechanism due to its incentive compatibility and budget-balance property.

6. Conclusion

This work proposes a truthful combinatorial double auction mechanism, TCMDAC, for cloud market. While considering the heterogeneity of cloud resources, TCMDAC models the users' demand and providers' offers by considering combinatorial bidding by users and divisible offers by providers. The model exploits the benefits of both double and combinatorial auctions which helps in preventing monopoly and enables competition at both side by considering the interest of both the users and providers. Resources are allocated to cloud users including the virtual user in order to maximize the total social welfare. After that the least efficient trades are removed and more eligible users are allowed to participate in the auction. Although this process loses some efficient allocations as compared to OPTIMAL mechanisms, yet the allocation leads to implement a truthful mechanism for the whole cloud market. After generating the feasible allocations, truthful payments for cloud users are obtained by critical price method. For each cloud user, critical payments are calculated which denotes the minimum values they can bid in order to win in the padded optimization problem. For cloud providers, marginal value based truthful payment schemes have been designed. Various auction properties of TCMDAC such as asymptotic efficiency, individual rationality, computationally feasibility, incentive compatibility, utilitarian social welfare, budget-balance are discussed and comparative study is done on the basis of these properties. The performance evaluation of TCMDAC indicates that the proposed model fits well for trading computing resources in the cloud market. Fig. 8 .

It is assumed that a provider does not violate its reported QoS which can be relaxed in the future work and certain mechanisms such as penalty or reputation can be applied in that scenario. Some more flexible bid offers such as discounted bid profiles for user, tiered bid profiles for provider etc. can be considered. Additionally, static bid profiles of users and provider are considered in this work which can be relaxed by considering Continuous Double Auctions (CDAs). In future, we aim to systematically consider these possibilities to observe their effect on cloud market functioning.

Acknowledgment

Authors would like to acknowledge the financial support of UGC UPE II, JNU, New Delhi to carry out this work.

References

Amazon;, 2016. EC2 Instances Info [WWW Document]. URL http://www.ec2instances.info/ (accessed 1.23.16).
AWS, 2016. Amazon EC2 Spot Instances [WWW Document]. Amaz. Web Serv. URL https://aws.amazon.com/ec2/spot/ (accessed 6.2.13).
Baranwal, G., Kumar, D., Raza, Z., Vidyarthi, D.P., 2017. A negotiation based dynamic pricing heuristic in cloud computing. Int. J. Grid Util. Comput 9, 83–96.
Baranwal, G., Vidyarthi, D.P., 2016. A truthful and fair multi-attribute combinatorial reverse auction for resource procurement in cloud computing. IEEE Trans. Serv. Comput. doi:10.1109/TSC.2016.2632719, 1–1.
Baranwal, G., Vidyarthi, D.P., 2015. A fair multi-attribute combinatorial double auction model for resource allocation in cloud computing. J. Syst. Softw. 108, 60–75. doi:10.1016/j.jss.2015.06.025.
Baranwal, G., Vidyarthi, D.P., 2014. An econometric based model for resource scarcity problem in cloud computing. In: Proceedings of the 2014 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT). IEEE, pp. 1–6. doi:10.1109/CONECCT.2014.6740276.
Bratton, K., Smith, V.L., Vannoni, M.G., Williams, A., 1982. Competitive market institutions: double auctions vs. sealed bid-offer auctions. Am. Econ. Rev. 72, 58–77. doi:10.2307/1808575.
Buyya, R., 2009. Market-oriented cloud computing: vision, hype, and reality of delivering computing as the 5th utility. In: Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGRID 2009, 25 doi:10.1109/CCGRID.2009.97.

Buyya, R., Abramson, D., Giddy, J., Stockinger, H., 2002. Economic models for resource management and scheduling in grid computing. Concurr. Comput. Pract. Exp. doi:10.1002/cpe.690.
Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A.F., Buyya, R., 2011. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw. – Pract. Exp. 41, 23–50. doi:10.1002/spe.995.
Cason, T.N., Friedman, D., 1996. Price formation in double auction markets. J. Econ. Dyn. Control 20, 1307–1337. doi:10.1016/0165-1889(95)00901-9.
Chichin, S., Vo, Q.B., Kowalczyk, R., 2015a. Double-sided market mechanism for trading cloud resources. In: Proceedings of the 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT). IEEE, pp. 198–205. doi:10.1109/WI-IAT.2015.85.
Chichin, S., Vo, Q.B., Kowalczyk, R., 2015b. Towards efficient greedy allocation schemes for double-sided cloud markets. In: Proceedings of the 2015 IEEE International Conference on Services Computing, SCC 2015, pp. 194–201. doi:10.1109/SCC.2015.35.
Chu, L.Y., 2009. Truthful bundle/multiunit double auctions. Manag. Sci. 55, 1184–1198. doi:10.1287/mnsc.1090.1015.
Chu, L.Y., Shen, Z.-J.M., 2008. Truthful double auction mechanisms. Oper. Res. 56, 102–120. doi:10.1287/opre.1070.0458.
Chu, L.Y., Shen, Z.-J.M., 2006. Agent competition double-auction mechanism. Manag. Sci. 52, 1215–1222. doi:10.1287/mnsc.1060.0528.
Chu, L.Y., Shen, Z.J.M., 2007. Trade reduction vs. multi-stage: a comparison of double auction design approaches. Eur. J. Oper. Res. 180, 677–691. doi:10.1016/j.ejor.2006.04.015.
Clarke, E.H., 1971. Multipart pricing of public goods. Public Choice 11, 17–33. doi:10.1007/BF01726210.
CloudHarmony, 2010. ECU [WWW Document]. CloudHarmony Blog. URL http://blog.cloudharmony.com/2010/05/what-is-ecu-cpu-benchmarking-in-Cloud.html (accessed 11.12.14).
cloudorado, 2016. cloudorado [WWW Document]. Cloud Comput. Comp. Engine. URL https://www.cloudorado.com/ (accessed 1.23.16).
de Vries, S., Vohra, R.V., 2003. Combinatorial auctions: a survey. INFORMS J. Comput. 15, 284–309. doi:10.1287/ijoc.15.3.284.16077.
Grosu, D., Das, A., 2004. Auction-based resource allocation protocols in grids. In: Proceedings of the 16th IASTED International Conference on Parallel and Distributed Computing and Systems, pp. 20–27.
Groves, T., 1973. Incentives in teams. Econometrica 41, 617–631. doi:10.2307/1914085.
Huang, D., He, B., Miao, C., 2014. A survey of resource management in multi-tier web applications. IEEE Commun. Surv. Tutor. 16, 1574–1590. doi:10.1109/SURV.2014.010814.00060.
Huang, P., Scheller-Wolf, A., Sycara, K., 2002. Design of a multi-unit double auction e-market. Comput. Intell. 18, 596–617. doi:10.1111/1467-8640.t01-1-00206.
Izakian, H., Abraham, A., Ladani, B.T., 2010. An auction method for resource allocation in computational grids. Futur. Gener. Comput. Syst. 26, 228–235.
Klemperer, P., 2004. Auctions: theory and practice, the toulouse lectures in economics [WWW Document]. SSRN Electron. J. doi:10.2139/ssrn.491563.
Kumar, D., Baranwal, G., Raza, Z., Vidyarthi, D.P., 2017. A systematic study of double auction mechanisms in cloud computing. J. Syst. Softw. 125, 234–255. doi:10.1016/j.jss.2016.12.009.
Lee, C., Wang, P., Niyato, D., 2015. A real-time group auction system for efficient allocation of cloud internet applications. IEEE Trans. Serv. Comput. 8, 251–268. doi:10.1109/TSC.2013.24.
Lehmann, D., O'Callaghan, L.I., 2002. Truth revelation in approximately efficient combinatorial auctions. J. ACM 49, 1–35. doi:10.1145/585265.585266.
Li, L., Liu, Y.-an, Liu, K.-ming, Ma, X.-lei, Yang, M., 2009. Pricing in combinatorial double auction-based grid allocation model. J. China Univ. Posts Telecommun. 16, 59–65. doi:10.1016/S1005-8885(08)60228-9.
Loertscher, S., Mezzetti, C., 2013. A Multi-Unit Dominant Strategy Double Auction.
Mashayekhy, L., Nejad, M.M., Grosu, D., 2015. A PTAS mechanism for provisioning and allocation of heterogeneous cloud resources. IEEE Trans. Parallel Distrib. Syst. 26, 2386–2399. doi:10.1109/TPDS.2014.2355228.
McAfee, R.P., 1992. A dominant strategy double auction. J. Econ. Theory 56, 434–450. doi:10.1016/0022-0531(92)90091-U.
Microsoft, 2016. Microsoft Azure [WWW Document]. Microsoft. URL https://azure.microsoft.com/en-in/ (accessed 2.23.13).
Myerson, R.B., 1981. Optimal auction design. Math. Oper. Res. 6, 58–73. doi:10.1287/moor.6.1.58.
Narahari, Y., 2014. Game Theory and Mechanism Design. World Scientific.
Nejad, M.M., Mashayekhy, L., Grosu, D., 2015. A family of truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in Clouds.pdf. IEEE Trans. Parallel Distrib. Syst. 26, 594–603. doi:10.1109/CLOUD.2013.14.
Pla, A., López, B., Murillo, J., 2015. Multi-dimensional fairness for auction-based resource allocation. Knowl.-Based Syst. 73, 134–148. doi:10.1016/j.knosys.2014.09.009.
Ray, A.K., Jenamani, M., Mohapatra, P.K.J., 2011. An efficient reverse auction mechanism for limited supplier base. Electron. Commer. Res. Appl. 10, 170–182. doi:10.1016/j.elerap.2009.11.002.
Rothkopf, M.H., Pekec, A., Harstad, R.M., 1998. Computationally manageable combinational auctions. Manag. Sci. 44, 1131–1147. doi:10.1287/mnsc.44.8.1131.
Samimi, P., Teimouri, Y., Mukhtar, M., 2016. A combinatorial double auction resource allocation model in cloud computing. Inf. Sci. 357, 201–216. doi:10.1016/j.ins.2014.02.008.
Stößer, J., Neumann, D., Weinhardt, C., 2010. Market-based pricing in grids: on strategic manipulation and computational cost. Eur. J. Oper. Res. 203, 464–475. doi:10.1016/j.ejor.2009.07.033.
Sun, J., Wang, X., Huang, M., 2013. An intelligent resource allocation mechanism in the cloud computing environment. In: Proceedings of the 2013 IEEE 3rd International Conference on Information Science and Technology, ICIST 2013, pp. 744–750. doi:10.1109/ICIST.2013.6747652.
Sun, Z., Zhu, Z., Chen, L., Xu, H., Huang, L., 2015. A combinatorial double auction mechanism for cloud resource group-buying. In: Proceedings of the 2014 IEEE 33rd International Performance Computing and Communications Conference, IPCCC 2014, pp. 1–8. doi:10.1109/PCCC.2014.7017104.
Urgaonkar, B., Pacifici, G., Shenoy, P., Spreitzer, M., Tantawi, A., 2007. Analytic modeling of multitier internet applications. ACM Trans. Web 1. 2 –es doi:10.1145/1232722.1232724.
Vickrey, W., 1961. Counterspeculation, auctions, and competitive sealed tenders. J. Financ. 16, 8–37. doi:10.1111/j.1540-6261.1961.tb02789.x.
Wang, X., Wang, X., Che, H., Li, K., Huang, M., Gao, C., 2015. An intelligent economic approach for dynamic resource allocation in cloud services. IEEE Trans. Cloud Comput. 3, 275–289. doi:10.1109/TCC.2015.2415776.
Wise, R., Morrison, D., 2000. Beyond the exchange–the future of B2B. Harv. Bus. Rev. 78, 86–96.
Wu, X., Liu, M., Dou, W.C., Gao, L., Yu, S., 2016. A scalable and automatic mechanism for resource allocation in self-organizing cloud. Peer-to-Peer Netw. Appl. 9, 28–41. doi:10.1007/s12083-014-0309-8.
Xia, M., Stallaert, J., Whinston, A.B., 2005. Solving the combinatorial double auction problem. Eur. J. Oper. Res. 164, 239–251. doi:10.1016/j.ejor.2003.11.018.
Xu, B., Zhao, C., Hu, E., Hu, B., 2011. Job scheduling algorithm based on Berger model in cloud environment. Adv. Eng. Softw. 42, 419–425. doi:10.1016/j.advengsoft.2011.03.007.
Zaman, S., Grosu, D., 2013. Combinatorial auction-based allocation of virtual machine instances in clouds. J. Parallel Distrib. Comput. 73, 495–508. doi:10.1016/j.jpdc.2012.12.006.
Zaman, S., Grosu, D., 2012. An online mechanism for dynamic VM provisioning and allocation in clouds. In: Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing, CLOUD, 2012, pp. 253–260. doi:10.1109/CLOUD.2012.26.

Dinesh Kumar is Ph.D. student at the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India. He completed his M.Tech in Computer Science from the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India. His research interests include resource provisioning and pricing in cloud computing.

Gaurav Baranwal is Assistant Professor in Department of Computer Science, Banaras Hindu University, Varanasi, UP, India. He did his M.Tech and Ph.D. in Computer Science from Jawaharlal Nehru University, India. He has published research papers in various peer reviewed International Journals (including IEEE, Elsevier, Springer, Wiley etc.) and in proceedings of various peer-reviewed conferences. His research interests include resource provisioning and service coordination in cloud computing and Internet of Things.

Zahid Raza is Associate Professor in the School of Computer and Systems Sciences, Jawaharlal Nehru University, India. He did his Ph.D. in Computer Science from Jawaharlal Nehru University, India. Prior to joining JNU, he served as a Lecturer in Banasthali Vidyapith University, Rajasthan, India. He has published many peer-reviewed articles and has proposed various scheduling models for computational grid, cloud and cluster systems. His research interests include parallel and distributed systems, evolutionary algorithms and multi-objective evolutionary algorithms.

Deo Prakash Vidyarthi is Professor in the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi. Dr. Vidyarthi has published around 90 research papers in various peer reviewed International Journals and Transactions (including IEEE, Elsevier, Springer, Wiley, World Scientific etc.) and around 50 research papers in proceedings of various peer-reviewed conferences in India and abroad. Dr. Vidyarthi has two books (research monograph) to his credit. One entitled “Technologies and Protocols for the Future Internet Design: Reinventing the Web” published by IGI-Global (USA) released in Feb. 2012, and another entitled “Scheduling in Distributed Computing Systems: Design, Analysis and Models” published by Springer, USA released in 2009. He also has contributed chapters in many edited books. He is in the editorial board and in the reviewer’s panel of many International Journals. Dr. Vidyarthi is the senior member of IEEE, International Association of Computer Science and Information Technology (IACSIT), Singapore, International Society of Research in Science and Technology (ISRST), USA and International Association of Engineers (IAENG). Research interest includes Parallel and Distributed System, Grid and Cloud Computing, Mobile Computing and Evolutionary Computing.