Simulation Modelling Practice and Theory 109 (2021) 102307

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

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

Fog-Integrated Cloud Architecture enabled multi-attribute combinatorial reverse auctioning framework

Anubha Aggarwal [a], Neetesh Kumar [b,*], Deo Prakash Vidyarthi [c], Rajkumar Buyya [d]

[a] Department of Computer Science & Engineering, Shri Mata Vaishno Devi University, Kakaryal, Katra, Jammu and Kashmir, 182320, India
[b] Department of Computer Science & Engineering, Indian Institute of Technology-Roorkee (IIT-R), 247667, India
[c] School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, 110067, India
[d] Cloud Computing and Distributed Systems (CloudS) Lab, School of Computing and Information Systems, The University of Melbourne, 3010, Australia

ARTICLE INFO

Keywords:
Fog computing
Resource procurement
Reverse auction
Fog service provider
Internet of Things (IoT)

ABSTRACT

Fog computing is an emerging service-oriented market in conjunction with Cloud computing to fulfill the resource demand of mobile users as well as IoT users for real-time applications. Auctioning in Fog computing is highly challenging due to mobility, dynamic pricing, real-time demand in comparison to Cloud based auctioning models. Further, due to users' mobility and limited Fog resources, existing reverse auction techniques developed for Cloud computing model cannot directly be applied for the resource procurement in Fog-Integrated Cloud Architecture (FICA). Therefore, a reverse auction-based model which includes customer, auctioneer, Fog provider, Cloud provider, and Fog & Cloud provider together as auction participants, is proposed in this work. The proposed model, for resource provisioning using a multi-attribute combinatorial reverse auction, is named as Fog-Integrated Cloud Auctioning Model (FICAM). FICAM pricing scheme includes three types of resources depending on their requirement i.e., local Fog, remote Fog, and Cloud. A truthful, robust, and fair algorithm for resource allocation is proposed considering response time, data source mobility requirements, and Fog resource limitations. To encourage providers to bid truthfully, the Vickrey model is extended. FICAM also introduces a new algorithm for resource procurement in which instead of giving all resources of the bundle, only the required resources at a time are given to the customer with the bundle discount. The discount is based on a certain threshold in the ratio of the availed amount of resources to the offered amount of resources. Rigorous experimentation exhibits that the proposed model offers a low resource procurement cost in polynomial time as compared to other state of the art algorithms.

1. Introduction

In recent, IoT has proliferated to create automated environments such as smart homes, healthcare, retail, transportation, security, surveillance, and other industrial necessities. Many of these applications have a rigid time-sensitive requirement for speedy processing of streamed data in order to initiate action in real-time. For example, in a home security system, on sensing a certain mishap a call is to be placed immediately to the police and the house owner. Nowadays, smart wearable, smart vehicles i.e., Internet of Vehicles (IoVs), mobile IoTs are in prevalence, and to make real-time decisions which require the computing resources closer to the data generation device.

* Corresponding author.
E-mail address: neetesh@cs.iitr.ac.in (N. Kumar).

https://doi.org/10.1016/j.simpat.2021.102307
Received 4 August 2020; Received in revised form 4 January 2021; Accepted 2 March 2021
Available online 10 March 2021

Fog computing is the best-fitted solution to handle such constraints of the requisite real-time computing [1]. Fog computing provides computing resources near to the IoT device with high network bandwidth and offers smooth Virtual Machine (VM) migration [2] as soon as the data source goes out of coverage area of the local Fog server. This allows for efficient communication between mobile objects and Fog resources. According to a research study [3], if Fog acts as an intermediate layer between the Cloud and consumer then not only the load on the Cloud reduces but the operational cost of the Cloud data centers also get reduced [4]. Cisco [5] stated that today's Cloud model is not suitable for the volume, variety, and velocity of data that IoT devices generate. Thus, there is a huge demand for nearby computing resources to handle such voluminous data for real-time computing generated by IoT applications [6]. The necessity of Fog computing for real-time computation with a good number of Fog service providers motivated us for proposing a model for resource procurement considering Fog-Integrated Cloud Architecture (FICA). FICA, being a commercial model, necessitates applying auction for the resource provisioning.

The reverse auction is an auction method in which the customer lists their requirements, and the service providers bid according to the customer's requirement. Thus, in the reverse auction, for the requirements of one customer, many service providers are ready to offer their services via quotes. Out of many available auction schemes such as fixed price, dynamic price, etc., the reverse auction is one that makes the market more competitive. It also helps in lowering the resource procurement cost [7], due to good competition, the customers are able to acquire the resources at the best price. Additionally, the e-procurement cost is an important parameter for the service provider in order to earn revenue as well as to survive in the competitive Cloud market [8]. For this work, a multi-attribute combinatorial reverse auction is chosen as it helps in comparing service providers not only on the price but many other attributes as well. According to a study, [9], some non-pricing attributes also play a significant role in a successful auction.

A reverse auction, for Cloud resource procurement, involves three participants: broker (also called the auctioneer), customer, and Cloud service provider. Customer submits its requirement in terms of a Call For Proposal (CFP) to a broker which invites several Cloud providers for the bidding. Broker declares winning Cloud provider based on some criteria and eventually allocates the resources of the winning provider to the customer. The Cloud computing reverse auction model is limited to the Cloud provider, customer, and broker as its vital components. However, in Fog computing, other members such as Fog provider exclusively and both Fog & Cloud provider together can also participate with varied needs. In general, the nature of Fog provider and customer's resource requirements are quite different due to the IoT functioning (real-time streamed data processing), limited Fog resources, and dynamic Fog pricing schemes. Thus, to successfully carry out a handshake between IoT and Fog computing technologies; a new architectural model with an efficient auction mechanism is warranted that can facilitate the customer, Fog provider, Cloud provider, and Fog & Cloud provider jointly. Customers should be flexible to use local or remote Fog resources in the FICA, whereas, pricing mechanisms in Fog and Cloud are different. Thus, an effective pricing scheme is needed for the FICA. Considering the above-mentioned necessities, a canonical reverse auction model designed for only Cloud computing scenario [10] cannot be applied directly to the three-layered FICA [11]. Therefore, in this work, an efficient auctioning model for FICA is proposed considering the above mentioned limitations. As per the author's best of knowledge, no reverse auctioning model for FICA is available in the state of the art. A Cloud auctioning model [10] is considered as the baseline model for this work. Novelties of the proposed Fog-Integrated Cloud Auctioning Model (FICAM), as compared to the baseline model, are as follows:

1. FICAM reduces the resource procurement cost as the number of resources are acquired as per the needs of the customer and not as per the offers by the provider. Thus, the proposed scheme overcomes the limitation of the baseline model where customers had to acquire the complete resource bundle that often may be more than the customer's requirement.
2. FICAM serves the time-sensitive requirements of the customer which is not considered in the baseline model developed for the Cloud-centric applications [10]. The customer can specify the expected response time in the requirement, based on this, services can be offered in the proposed FICAM.
3. FICAM supports collaboration between Fog and Cloud providers in order to enable IoT oriented services (in real-time) and improves the quality of experience (QoE). On other hand, this feature is not supported by the base model.
4. FICAM allows billing by accounting dynamic pricing scheme that is not considered in the baseline model.
5. With FICAM, customers experience reduced latency as it avoids the network delay, and the resources are available at the edge. Hence, it improves the quality of services (QoS).
6. FICAM accounts the data source mobility, to the best of the authors' knowledge, it has not been considered by any available Cloud resource auctioning model.

In order to offer the above mentioned features by the FICAM, authors significant contributions are itemized as follows.

1. A multi-attribute combinatorial reverse auction model is proposed for the resource procurement considering FICA with four entities: broker, customer, Fog service provider, and Cloud service provider as the essential components. Broker acts as a middleman between service provider and customer, and the customer is the user of the Cloud/Fog services. Cloud and Fog service provider offer available Cloud and Fog resources respectively.
2. For the services, customer prepares a list of the requirements in the form of a CFP to the broker. On the other hand, Fog, Cloud, Fog & Cloud service provider give their quotations in the form of a bid to the broker. Based on the quotations, the broker determines the winner who provides their services to the customer.
3. The proposed pricing scheme (Fig. 2) is based on three types of pricing: local, remote, and Cloud pricing. The service provider charges local and remote prices if allocated VM is on the Fog node. The local price of the VM is charged when the data source is local but as soon as the data source migrates, the user is charged with the remote price for the respective VM. A user is charged with the Cloud price of a VM if the allocated VM is on the Cloud.

4. The Vickrey auction [12,13] algorithm is extended via an incentive approach, it offers incentives to the provider for truthful bidding, and it also applies a penalty on untruthful bidding. Further, the proposed multi-attribute combinatorial reverse auction algorithm is improved to run only in polynomial time as the worst-case time complexity.
5. The proposed algorithm also determines the multiple winners, and the respective quantity of VMs is availed from each winner provider's bundle. Considering this scenario, the final billing is done on the basis of threshold, bundle discount, and ratio of the number of used resources to the offered resources.
6. The performance evaluation of the FICAM is done under eight different scenarios. The comparative results are evidence for the effectiveness of the model for all the scenarios considering resource procurement cost, and by varying several performance metrics. As an outcome, FICAM significantly reduces the overall resource procurement cost over the baseline model.

The rest of the paper is organized as follows. Section 2 briefs the recent related work on the reverse auction. Section 3 describes various modules of the proposed FICAM model along-with the algorithms. Section 4 details the experiments carried out for the performance analysis of the model. Finally, Section 5 concludes the work by pointing out some future research work.

2. Related work and background

This section puts forth some related work besides the baseline work which is the motivation of this work.

Luan et al. [14] discussed the system architecture of Fog computing and compared it with Cloud computing on its design and research issues. Sarkar et al. [15] explored the opportunities of Fog computing in IoT. Through experiments, they proved that if Fog computing is used with Cloud computing, it is a better option for a good number of high latency applications. Yi et al. [2] demonstrated the concept of VM migration in Fog computing. They also discussed three-layer architecture goals like latency, efficiency, and generality; challenges such as choice of virtualization technology; resource provisioning; and applications like smart home, smart grid, the vehicle of Fog computing. Fan et al. [16] proposed a load balancing scheme in Fog computing to reduce the latency. They Shaw that both communication and computation load should be balanced in order to minimize the response time. From these studies, it is inferred that Fog computing is enormously useful for the implementation of future IoT applications, particularly, real-time applications.

Jiao et al. [17] proposed a fair, rational auction-based model for resource procurement in public blockchain networks. They proposed constant and multi demand bidding schemes. Experimental results confirmed the maximization of social welfare in blockchain networks through the use of bidding schemes. However, this model is suitable for an architecture where there are multiple miners (bidders) and one Fog/Cloud provider. Further, Mazin et al. [18] proposed a decentralized, transparent, secure reverse bidding scheme developed using the key feature of blockchain & smart-contracts. The bidding process is started by the request for service from the users or devices. Services are provided by nearby fog devices and these devices make bid offers in return for service. This scheme imposes a penalty on those who do not participate fairly in auction and also integrates a reputation system. However, this model does not consider a combinatorial auction. In combinatorial auction, as the buyer will purchase a combination of items rather than individual items which leads the buyer to have some profit.

Song et al. [19] introduced a combinatorial reverse auction scheme in which many providers tie-up to fulfill customer's demand for the best pricing. This enhances their chances of winning. Vries et al. [20] argue that if instead of a single item, bidding is done on a group of items (bundle), resource procurement cost is reduced. Prasad et al. [21] addressed the issue, arising out of procuring multiple resources from several Cloud vendors using an auction. Experimental results exhibit that a combinatorial auction is superior to a single auction. Liwang et at. [22] proposed a novel computation offloading marketplace in vehicular networks where a VCG-based reverse auction mechanism utilizes integer linear programming (ILP) while satisfying the desirable economical properties of truthfulness and individual rationality. They developed an efficient unilateral-matching-based mechanism with polynomial computational complexity, truthfulness, and individual rationality properties as well as matching stability. The lacking of the model is that it does not consider combinatorial, multi-attribute auction i.e., the only price is considered as the winning parameter.

Most of the discussed work consider the price as an attribute to win the auction. However, many QoS parameters, i.e., non-price attributes are equally important in order to determine the winners. Pla et al. [23] introduced a multi-attribute reverse auction mechanism, VMA2, and classified three types of attributes: verifiable attributes, auctioneer provided attributes and non-verifiable attributes. They stated that verifiable attributes and auctioneer provided attributes to ensure truthfulness and trust. This mechanism gives incentive to the truthful bidders. VMA2 was compared with the unattributed auction method (auction that considers the price as the only attribute) and it was found that its resource procurement cost was less than that of unattributed auctions. VMA2 encouraged bidders to bid their true value, but if only the providers who offer cheaper prices and high-quality services, they win, and other providers lose interest and leave the auction market. Due to this, a bunch of service providers start controlling the market. This is known as bidder drop out problem [24]. An auction mechanism should be fair to all the service providers.

Gaurav et al. [10] introduced a fair reverse auction mechanism (TFMCRA) for resource procurement in Cloud computing. It is a multi-attribute, combinatorial, and truthful reverse auction mechanism. For encouraging providers to bid truthfully, TFMCRA uses Vickrey payment. It considers price as well as non-price attributes for the winner determination. It is a truthful, fair, multi-attribute, combinatorial reverse auction model for resource procurement in Cloud computing that executes in polynomial time. However, this model only caters to Cloud computing. Nonetheless, this work forms a baseline for the proposed resource provisioning model considering FICA. Fog architecture is a three-tier architecture and is different from Cloud architecture. Pricing schemes of Fog computing are also different as the types and demands of Fog resources are different. Further, data source mobility issue [25] appears in Fog and not in the Cloud, therefore it is not addressed in the baseline model [10]. Thus, a need arises for designing a new resource procurement mechanism for the three-tier Fog-integrated cloud architecture model.

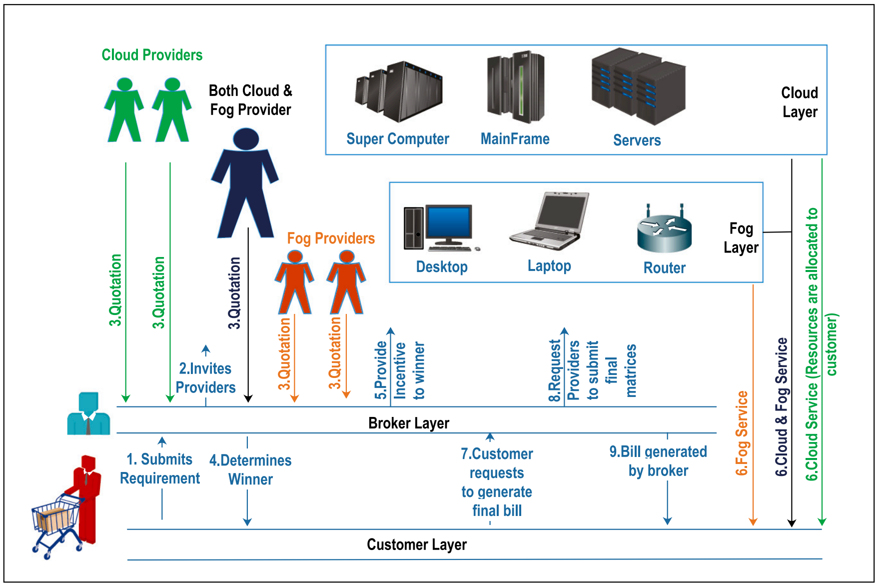


Fig. 1. Reverse auctioning in the proposed FICA.

3. The proposed model: FICAM

This section details about the proposed FICAM model including its architecture, pricing, and resource procurement algorithm for three-tier computing architecture. FICAM differentiates between Fog and Cloud computing especially on data source mobility, location awareness, and dynamic pricing schemes based on real-time workload.

3.1. The proposed reverse auction architecture

Fig. 1 shows three-tier architecture of the proposed FICAM model. FICAM ensures its customers about the services at the best price considering multiple desired QoS parameters. The participants, in the proposed auction-based model, are Customer, Broker, Fog Provider, Cloud Provider, and Fog & Cloud Provider.

Broker:- Broker is a middleman between the customer and service provider (Fog, Cloud, Fog & Cloud) [10]. A broker, on receiving the requirements from a customer, invites service providers to submit their bids. Based on the quotes, some historical facts, and customer requirements, the broker determines the winners. In the process, the broker may impose some penalty on those providers who indulge in cheating. It compensates customers. The broker also offers some incentives to the winning providers for biding truthfully.

Customer:- Customer can be an individual or an organization. A customer has two types of requirements: i. Real-time/time-sensitive requirement ii. Normal requirement. Various IoT applications, gaming applications, smart vehicles, etc. exhibit time-sensitive requirements. A customer submits the application and specifies the response time requirements to the broker.

Cloud Provider (CP):- CP provides Cloud computing resources on rent at the Cloud layer. The customer's requirement is verified; if it is normal and CP can offer the demanded resources, then it surely takes part in the bidding. If the customer's requirement is time-sensitive, then CP may or may not participate. CPs can participate only if they are able to satisfy the expected response time specifications from the customer.

Fog Provider (FP):- FP provides Fog services/resources for rent at the Fog layer. If the customer's requirement is time-sensitive, and FP is able to meet so, it takes part. Fog devices are located in close proximity to users and usually responsible for intermediate computation and storage [26]. On the contrary, if the customer's requirement is normal then FPs may or may not participate because Fog devices have relatively limited power and computation resources compared to Cloud computing [27]. It depends upon the current workload on the Fog devices and the FP's profit.

Fog & Cloud Provider (FCP):- FCP offers services for both Fog and Cloud resources. FCP owns both Fog and Cloud resources. Sometimes, the customer may need big configuration machines for the processing which cannot be satisfied by the Fog services, or the customer needs a quick response that cannot be met by the Cloud provider. To serve the customer with better QoS, FCP conceived that he is equipped with both Fog and Cloud resources. Service provider either may own both Fog and Cloud resources or may tie-up with other providers to offer both Fog and Cloud resources. As FCP provides both Fog and Cloud resources, he can participate in bidding for any type of customer's requirement.

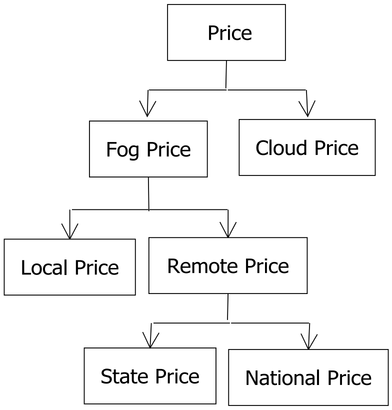


Fig. 2. Pricing schemes used by FICAM.

3.2. Fog pricing schemes

Fig. 2 shows the possible pricing types based on the computing requirements. To deal with both Fog and Cloud resources, pricing schemes are categorized into Fog price and Cloud price. Technically, Fog resources are meant for time-sensitive applications, therefore, meeting time deadline is a necessity. Therefore, to meet the deadline, when the data source is mobile, VM migration [2] takes place in Fog computing. As IoT devices move, their VMs are migrated to the nearest Fog center closer to their current location. This involves some VM migration costs. At any point of the time, load varies on different Fog centers and therefore, due to dynamic pricing, pricing of the same VM may vary with different Fog providers at different times and places. Keeping VM migration and dynamic Fog pricing in view, usually, Fog pricing is divided into two types: local and remote. In FICAM, each provider indicates three types of prices for a VM in their quote. These are Local Price, Remote Price, and Cloud Price as deliberated in the subsequent section.

Local Price :- The allocated Fog resources are registered in a Fog center near to the data source. The area served by the Fog center is called base area. The customer is charged according to some local price rate of the Fog resources as long as the data source remains in the base area.

Remote Price:- Remote prices are applicable only when the data source moves from the base area to some other area. This necessitates the VM migration from the Fog center/server of the base area to another Fog center in the migrated area. This onward, the customer is charged according to the remote price rates of the Fog nodes. The remote price depends on the range of mobility of the data source. Based on the range of data source mobility, the remote price is further classified into two types: state remote price and national remote price. This division is inspired by the types of roaming in telecommunication, however, it may be given some other names [28]. The objective of such division is to specify the rules that are applicable to the area in which the Fog centers are registered.

(a) State Remote Price:-In this scheme, the limit of mobility of the data source is in the state of the registered Fog center. Consider a scenario in which a customer is using the smart vehicle which uses the Fog resources. Eventually, smart vehicle crosses the district range but still within the state boundaries. Provider charges the customer as per the state remote price for the acquired Fog resources.

(b) National Remote Price:- In this scheme, the data source's mobility limit is the entire country. The data source may roam anywhere in the country, and VM is migrated to the closest Fog center according to its availability.

Cloud Price:- If VMs are allocated on the Cloud resources, the pricing for the respective resources is done at the Cloud price rate of VMs. Depending upon the resources, availed by the customers, pricing is done. If a provider offers only Cloud resources, or only Fog resources, or mixed Cloud & Fog resources, the customer is charged accordingly.

3.3. The algorithm

An algorithm is developed for resource procurement considering FICA using reverse auction. For making algorithm incentive-compatible, an approach similar to [10] applying Vickrey auction [12] with some special properties i.e., data source mobility and resource limitation in FICAM is proposed. The methodology, of the proposed algorithm, is summarized in the following steps.

1. The CFP in the form of requirement vector is submitted to the broker by the customer.
2. The broker invites the providers to submit their quotes (bids) that satisfies the customer's requirements.
3. Bidding is processed, completed and closed.
4. Total bundle price ($tlp + trp + tcp$), of each provider, is estimated by the broker.
5. Since, the history of provider matters a lot, history attributes of the provider are added in the quotation by the broker. It is now referred to as the extended Quotation.

Table 1
Notation.

Notation	Description	Notation	Description
mcf	Mach configuration	*qr*	Quantity required
qo	Quantity offered	*qt*	Quantity taken
et	Expected time	*resp_time*	Response time
pb_l	Local probability	pb_r	Remote probability
lm	Data source limit mobility	*quot_vec*	Quotation vector
lp	Initial local price matrix	*rp*	Initial remote price matrix
cp	Initial Cloud price matrix	*tlp*	Total local price
trp	Total remote price	*tcp*	Total Cloud price
repu	Reputation	*pri*	Priority
ctq	Current taken quantity	*req*	Requirement
discob	Discount on bundle	*ebp*	Estimated bundle price vector
flt	Final local time matrix	*frt*	Final remote time matrix
fct	Final cloud time matrix	*fqt*	Final quantity taken matrix
fb	Final bill Vector	*extquot_vec*	Extended Quotation vector
P	Provider vector	*sorted_provider*	Sorted provider list
rw	Runner up winner	*w*	Winner
w_vec	Weight vector	QoS_len	Length of QoS vector
w_repu	Weight reputation	w_pri	Weight priority
M	Total no of resources with multiplication of their attributes	$for\_incentive$	If a module is called in order to give incentive to winning providers.
point list	List of evaluated value of all providers based on non-price attributes.	*score list*	List of scores of all providers based on value of non-price attributes.

6. Winner and Final quantity of VMs taken (i.e., fqt vector of each VM type from each winner provider) are determined by the broker.
7. The incentive is offered to the winner, by the broker, for truthful bidding.
8. Resources are allocated to the customer.
9. The customer requests the broker to generate the final bill and consequently, the broker requests the customer for the feedback on delivered QoS.
10. Broker request the provider to submit final local time matrix (flt), final remote time matrix (frt), and final Cloud time matrix (fct).
11. Finally, the bill is generated by the broker for each winning provider, and the same is conveyed to the customer.

The rest of the algorithmic details are presented as follows.

3.3.1. Input

The algorithm takes the following input from the customer.

Machine Configuration Vector (mcf):- It contains the configuration of machines that a customer needs. This vector is denoted by $(mcf^1, mcf^2, \ldots, mcf^n)$.

Quantity Required Vector (qr):- It contains the quantity of each type of required machine. It is denoted by (qr^1, qr^2, $\ldots, qr^n$), where, qr^i corresponds to quantity of the instance of mach configuration i.e., mcf^i required by the ith customer.

Expected Time Vector (et) :- It contains the estimated time usage of each machines. It is denoted by (et^1, $et^2, \ldots, et^n$), where et^i is the expected time usage of machine mcf^i.

Response Time Vector (res_time):- The response time res_time^i is the expected response time from each machine mcf^i for real-time applications. It is denoted by $(res\_time^1, res\_time^2, \ldots, res\_time^n)$.

Local and Remote Probability (pb_l, pb_r):- The response time is also dependent on the distance of the application from the data source, especially if the customer is utilizing resources for the time-sensitive applications. Consequently, the local and remote probability of the data source location is required. The probability of locally used resources is pb_l, and the probability of remotely used resources is pb_r.

Limit of Mobility (lm):- It denotes the extent, a data source can be expected to be mobile. The limit may be state or national. If the limit is national it means a data source can move within the national boundaries denoted by lm =state/national. It is to make sure that despite the mobility of the data source, the response time (deadline) for real-time applications need to be met. For a non-real-time application, response time does not matter. Therefore, VM migration does not take place even if the resources are allocated to Fog and data source moves. In this case, $pb_l = 1$, $pb_r = 0$ and lm = 'local'.

QoS Vector and Weight Vector (QoS_vec and w_vec):- These vectors consist of the value of non-price attributes. The length of this vector is denoted by QoS_len. Weight, given by the customer to these QoS, is denoted as $weight\_vec$. From a provider, say p_x, quotation is taken as the input which consists of the following:

Initial Remote Price Vector (rp_x):- This vector contains remote price of each configured machine according to the data source limit mobility i.e., lm specified in the customer's requirement. It is denoted by $(rp_x^1, rp_x^2, \ldots, rp_x^n)$ where rp_x^i corresponds to the remote price of mcf^i provided by p_x. Value of rp_x^i is zero, if p_x provides mcf^i on Cloud layer or if the data source is local i.e., $pb_r = 0$.

Initial Cloud Price Vector(cp_x):- This vector contains Cloud price of each configuration machine. It is denoted by ($cp_x^1, cp_x^2, \ldots, cp_x^n$), where cp_x^i corresponds to the Cloud price of mcf^i provided by p_x. The value of cp_x^i is zero if p_x is providing mcf^i on the Fog layer.

Quantity Offered Vector (qo_x):- This denotes the quantity of each machine configuration, a provider is able to offer. It is denoted by ($qo_x^1, qo\ _x^2, \ldots, qo_x^n$), where qo_x^i is the quantity of i_{th} machine configuration that provider p_x is offering.

Discount on Bundle ($discob_x$):- As obvious, the customer is eligible to get the discount on a bundle. A discount i.e., $discob_x$ is the amount that provider i.e., p_x offers to the customer on the final bill.

QoS Vector:- It contains the threshold value of the non-price attribute (delay time) that customer wants from Cloud and Fog services. If an attribute is highly desirable then the threshold value is minimum, otherwise, it is maximum. The model asks for the desired QoS from the customer as well as from the provider. The customer gives its QoS requirement as the quality expected. The provider (through customer QoS) will display the values of those non-price attributes that it is able to supply. The provider may offer better QoS than the demanded one. At the time of final price settlement, the broker compares the delivered QoS with the promised QoS by the provider to ascertain if the provider cheated with the customer.

It is advantageous to consider the price of each VM rather than the price of the whole bundle. It is because the proposed method first tries to avail the cheaper resources from the providers before giving the opportunity to some other providers. Therefore, the quantity of the availed VMs from each provider may not be the same as the quantity of the offered VMs by a provider in the quote. Though, the algorithm considers the probability of the data source being local and remote, the actual period of time for which the data source will be local or remote can be determined only at the end, i.e., after the customer releases all the consumed resources. If the algorithm takes the bundled price from the provider then only the provider will be able to calculate the final bill, and no way a customer can verify the bill. From the bundle price, it is difficult to estimate the final bill according to the final quantity of the resource matrix & actual time taken matrix in a combinatorial auction. Thus, in order to prevent a customer from cheating, it is required to have the pricing rates and actual execution time of each VM type.

Broker, not only considers the provider's quotation but also the attributes based on the past experience such as reputation and priority. Calculation of these attributes is not dependent on whether resource procurement is for integrated Fog & Cloud architecture or for Cloud alone. Therefore, the calculation of reputation and priority is done in the same way as in [10].

3.3.2. The algorithm

The algorithm for the FICAM model includes five modules (Algorithms 2–6). It is "Estimated Bundle Price Algorithm" to estimate the bundle price of each provider, "Sort Providers Algorithm" to sort the providers on the basis of the attributes; "Winner & Quantity Determination Algorithm" to find the winners and the quantity of each VM types to avail from each provider, "Incentivize Winner Providers Algorithm" to make the model truthful; and "Price Settlement algorithm" to calculate final bills and impose a penalty on the fraud providers if any. Notation, used in the algorithms, are listed in Table 1. The time complexity of the algorithm is given in Appendix A.

Algorithm 1: FICAM

```
Customer submits its requirement to the broker
Broker starts auction
    ▷ Each provider submits quotation to the broker
for each p_x ∈ P  do
    quot_vec_x=(lp_x, rp_x, cp_x, qo_x, discob_x)
end for
                    ▷ Broker extends Quotation and calculate each provider's estimated bundle price
for all p_x ∈ to P do
    extquot_vec_x = (quot_vec_x, repu, pri)
end for
ebp = EstimatedBundlePriceCalculator (quot_vec, req)
    ▷ Broker decides winners
(Winners, fqt, point list, score list) = Winner & Quantity Determination (extquot_vec, ebp, req, false)
(Incentive)=Incentivize Winning Providers (Winners, extquot_vec, req, point list, score list)
Customer use resources
    ▷ Each winning provider p_x submits final time taken matrices
for all p_x ∈ Winners do
    Submit flt_x, frt_x, fct_x
end for
    ▷ Broker calculates final bill and apply penalty to cheater providers
fb=PriceSettlement(Winners, Incentive, fqt, flt, frt, fct, extquot_vec)
Output: fb
```

First, in the submitted quotation of the provider, attributes like reputation and priority are extended by the broker. Then, the broker calculates the estimated bundle price for each provider using the Estimated Bundle Price Calculator (Algorithm 2).

Estimated Bundle Price Calculator: This module calculates the total price of the bundle for each provider. It takes care of local and remote probabilities and also the mobility limitations of the data source. Initially, for bidding purposes, the local and remote probability is taken from the customer but during final settlement, the bill is calculated by the provider based on the actual local, remote, Cloud time taken matrices. For calculating the estimated bundle price of the provider i.e., p_x, it calculates total local price (tlp), total remote price (trp), and total cloud price (tcp). This estimated bundle price helps the broker in comparing the providers. $tlp+trp$ is the total per-minute price of the Fog resources, and tcp is the total per-minute price of the Cloud resources. Total estimated bundle price of a provider p_x is given by Eq. (1) where $discob_x$ is discount on the bundle which provider offers.

$$ebp_{\mathrm{x}} = (tlp + trp + tcp) \times \frac{(100 - discob_{\mathrm{x}})}{100} \tag{1}$$

Algorithm 2: Estimated Bundle Price Calculator

Input: quot_vec, req
for all $p_x \in$ P **do**
  localPriceVec= quot_vec$_{\mathrm{x}}$.lp$_{\mathrm{x}}$
  remotePriceVec = quot_vec$_{\mathrm{x}}$.rp$_{\mathrm{x}}$
  CloudPriceVec= quot_vec$_{\mathrm{x}}$.cp$_{\mathrm{x}}$
  tlp $=\sum_{i=1}^{n}(localPriceVec(i) \times pb_l \times qo^{\mathrm{i}}{}_{\mathrm{x}} \times et^{\mathrm{i}})$
  trp $=\sum_{i=1}^{n}(remotePriceVec(i) \times pb_r \times qo^{\mathrm{i}}{}_{\mathrm{x}} \times et^{\mathrm{i}})$
  tcp $= \sum_{i=1}^{n}(CloudPriceVec(i) \times qo^{\mathrm{i}}{}_{\mathrm{x}} \times et^{\mathrm{i}}$
  ebp$_{\mathrm{x}}$=(tlp + trp + tcp) × (100 - discob$_{\mathrm{x}}$)/100
end for
Output: ebp

The broker now determines the winners and also the quantity of each VM type to avail from each winner. It is not possible that every provider can satisfy total requirements of the customer. However, several providers together can become winners to meet the overall customer requirements. Once the customer's requirement is satisfied, the winner determination algorithm stops.

Algorithm 3: Sort Providers

Input: extquot_vec, req, ebp, for_incentive
for all $p_x \in$ P **do**
  M(x) = $\sum_{i=1}^{n} mcf^{\ \mathrm{i}} \times qo^{\ \mathrm{i}}{}_{\mathrm{x}}$
  BidDensity(x) = $ebp_{\mathrm{x}}/\sqrt{M(x)}$

$$score(x) = \sum_{i=1}^{QoS\_len} QoS(i) \times w\_vec(i) + repu \times w\_repu + priority \times w\_pri \tag{2}$$

$$point(x) = BidDensity(x) \times 1/score(x) \tag{3}$$

end for
Sort providers on the basis of points (ascending order)
if for_incentive == false **then**
  Output: point list, score list, sorted_provider
else
  Output: sorted_provider
end if

Winner and Quantity Taken Matrix Determination: In order to determine the winners, sorting of the providers is done. For this, $\sqrt{M}$ approximation technique is used as done in the baseline model [10]. Lehmann et al. [29] suggested that using $\sqrt{M}$ approximation, we can change the time complexity from exponential to polynomial. In the baseline model, the quantity of the resources of a VM type availed from the provider is the same as offered by the provider. FICAM considers the availed quantity of resources of VMs according to the customer's requirement.

The providers offering at least one unit of any VM type are declared as winners. The unit of resources of each VM type taken from the winning providers is stored in final quantity taken matrix (fqt). To begin with, resources are taken from the first provider in the sorted list of providers, if the provider is able to offer the desired quantity of all VM types demanded by the customer. In that case, the first provider is the sole winner. If not so, the remaining quantity of the resources is taken from the second provider in the sorted list of providers. The process stops if customer request is full-filled or providers are unable to satisfy the customer requests (if there is no VM type availed from a service provider p_x, i.e., $\sum_{i=1}^{n} qt^{\mathrm{i}}{}_{\mathrm{x}} = 0$, where, i signifies the VM type).

Algorithm 4: Winner & Quantity Determination

Input: extquot_vec, req, ebp, for_incentive
if for_incentive == false **then**
    sorted_provider, point list, score list = Sort_Providers (extquot_vec, req, ebp, for_incentive)
else
    sorted_provider = Sort_Providers(extquot_vec, req, ebp, for_incentive)
end if
▷ calculate winners and how much quantity of each machine to take from them
for each p_x ∈ to sorted_provider **do**
    qo_x = $quot\_vec_x.qo_x$
    for i=1 to n **do**
        if $ctq^i < qr^i$ **then**
            if $qo^i{}_x >= qr^i - ctq^i$ **then**
                $qt^i{}_x = qr^i - ctq^i$
            else
                $qt^i{}_x = qo^i{}_x$
            end if
            $ctq^i \mathrel{+}= qt^i{}_x$
        end if
    end for
▷ once all resources in desired quantity have been taken winner determination is complete
    if $\sum_{i=1}^{n} qt^i{}_x = 0$ **then**
        break
    else
        if for_incentive==false **then**
            $fqt_x = qt_x$
        end if
        Winners=Winners ∪ provider
    end if
end for
if for_incentive == false **then**
    Output: Winners, fqt, point list, score list
else
    Output: Winners
end if

Algorithm 5: Incentive to Winning Providers

Input: Winners, extquot_vec, req, point list, score list
for all p_x ∈ Winners **do**
    $M(x) = \sum_{i=1}^{n} mcf^{\,i} \times qt^{\,i}{}_x$
    rw=-1
    (Winners′) = WinnerDetermination(req, $extquot\_vec_{-x}$, ebp_{-x}, true)
▷ Here Winners′ vec is traversed from last to first
    for all p_y ∈ Winners′ **do**
        if $p_y \notin$ Winners **then**
            $rw \leftarrow p_y$
            break
        end if
    end for
    if $rw > 0$ **then**
        Incentive_point(x)= (point(rw) - point(x))
    else
        Incentive_point(x)=0
    end if
    Incentive(x) = Incentive_point(x)× $\sqrt{(M(x) \times score(x)}$
end for
for $p_x \notin$ Winners **do**
    Incentive(x)=0
end for
Output: Incentive

Incentive to Winning Providers: This module extends Vickrey auction [12] scheme for the winning providers, in order to motivate them to bid truth value. This scheme makes FICAM incentive compatible. In the Vickrey auction, a customer need not pay the winner (w) according to the bidding of the winner. Alternatively, the customer pays according to the runner-up provider (rw) which is the winner if w does not participate in the auction. When there are two or more candidates for rw, the runner-up winner is the one whose bidding is higher.

However, the winners cannot be paid directly the estimated bundle price of rw as it is based on the quantity of VMs offered by the provider and not on the basis of the final quantity of the VMs availed from the provider. Additionally, the winner cannot be directly paid the final price of the runner-up winner because the final price of the runner-up winner is zero (as the final quantity taken from rw is zero). Therefore, instead of directly applying Vickrey pricing as in the base model TFMCRA, FICAM gives incentive to the winner based on the quotation of the w and rw.

For the uni-attribute auction, the incentive of the winner is the price difference between runner-up winner and winner as shown in Eq. (4).

$$Incentive(w) = price(rw) - price(w) \tag{4}$$

The proposed incentive scheme uses multi-attribute auction, where the overall price is determined considering the overall attributes and not just the price. Here, a winner is paid according to the evaluated value (i.e., point) of runner-up winner(rw) and winner(w). For obtaining that value, the incentive point is calculated which is the difference between the point of runner-up winner and winner as shown in Eq. (5).

$$Incentive\_point(w) = point(rw) - point(w) \tag{5}$$

After calculating the incentive point, incentive price needs to be extracted. Since, point is calculated using Eq. (3) Algorithm 3, therefore, price can be extracted from the point using Eq. (6).

$$Incentive(w) = Incentive\_point(w) \times \sqrt{(}M(w) \times score(w)) \tag{6}$$

Score is calculated using Eq. (2) and M(w) is calculated using Eq. (7)

$$M(w) = \sum_{i=1}^{n} mcf^{i} \times fqt_{w}^{i} \tag{7}$$

Price Settlement Algorithm This algorithm generates the final bill vector for each winning provider. Customer reports to the broker about the delivered quality of the resources. If the delivered QoS is higher than the proposed QoS, it works fine. In case, it is less than the proposed QoS, it indicates a cheat on the provider's part. Considering this, the score is recalculated. If the recalculated score is equal to the initial score then an incentive is given to the provider. But if the recalculated score is less than the initial score, i.e., if the provider cheats, then no incentive is given to the provider. If provider cheats, the final bill is produced in proportion to $\frac{final\_score}{initial\_score}$ as a compensation to the customer and as a penalty to the provider. Thus, if provider cheats, then $final\_score$ is less than $initial\_score$. In this, $initial\_score$ is calculated at the time of sorting the providers, and $final\_score$ is calculated after the customer reporting on the delivered QoS.

Algorithm 6: Price Settlement

Input: Winners, Incentive, fqt, flt, frt, fct, extquot_vec
Customer tells broker delivered QoS of winning providers
Broker recalculates score (final_score) of winning providers
for each $p_x \in$ Winners **do**
  for i=1 to n **do**
    bill += ($\text{lp}^{\text{i}}{}_{\text{x}} \times \text{flt}^{\text{i}}{}_{\text{x}} + \text{rp}^{\text{i}}{}_{\text{x}} \times \text{frt}^{\text{i}}{}_{\text{x}} + \text{cp}^{\text{i}}{}_{\text{x}} \times \text{fct}^{\text{i}}{}_{\text{x}}) \times \text{fqt}^{\text{i}}{}_{\text{x}}$
  end for
  ▷ if customer has availed at least 50% resources of what provider has offered then only customer gets discount on bundle and that too in proportion
  ratio= $\frac{\sum_{i=1}^{n} fqt^{\text{i}}{}_{\text{x}}}{\sum_{i=1}^{n} qo^{\text{i}}{}_{\text{x}}}$
  if ratio >= r_{x} **then**
    bill = bill × (100 - (discob_{x} × ratio))/100
  end if
  if final_score (x) is >= score(x) **then**
    fb_{x} = bill + Incentive(x)
  else
    fb_{x} = bill × final_score / initial_score
  end if
end for
Output: fb

Bill is calculated based on initial local, remote, Cloud price matrices; final quantity taken matrix; and final local, remote, Cloud time taken matrices. Before providing the discount, the ratio of quantity is calculated using Eq. (8) which is the ratio of quantity availed and quantity offered by the provider. If $ratio > r_{\text{x}}$, a discount is given where r_{x} is decided by the provider. In this work, for simplicity in the calculation, the value of r_{x} is considered as 0.5.

$$\text{ratio} = \frac{\sum_{i=1}^{n} fqt^{i}{}_{\text{x}}}{\sum_{i=1}^{n} qo^{i}{}_{\text{x}}} \tag{8}$$

FICAM satisfies certain auction properties like incentive-compatible/truthful, budget balanced, individually rational, etc. Proof of properties, i.e., truthfulness, monotone, non-dominant, robust, egalitarian social welfare (fairness), the budget balanced, and individual rationale is given in Appendix B. We have compared FICAM with other models and a summary of comparison is given in

Table 2
Comparative study of various state of arts.

Features	Reverse auctioning model						
	FICAM	TFMCRA	Prasad	C-DISC	Modica	BDRBFC	Liwang
Combinatorial auction	✓	✓	✓	✗	✓	✗	✗
Multi-attribute auction	✓	✓	✓	✓	✗	✓	✗
Budget balanced	✓	✓	✓	✓	–	–	✗
Robust	✓	✓	✗	✗	✗	✓	–
Bidder’s optimality	✗	✗	✓	✗	✓	–	–
Non-dominant	✓	✓	–	✗	✓	–	–
Incentive compatible	✓	✓	✗	✓	✗	✓	✓
Egalitarian social welfare	✓	✓	✗	✗	✗	✓	✗
Cloud architecture applicable	✓	✓	✓	✓	✓	✓	✗
FICA applicable	✓	✗	✗	✗	✗	✓	✗
Satisfy mobile data source/IOT need	✓	✗	✗	✗	✗	✓	✓

TFMCRA - [10] Prasad - [21] C-DISC - [35] Modica - [36] BDRBFC - [18] Liwang - [22].

Table 2. Comparison of FICAM with other reverse auctioning models is based on reverse auction properties such as whether model supports; combinatorial auction — where buyers bids on combination of items rather than on individual items [30], multi-attribute auction — where attributes other than price are also considered during auctioning [31], budget balanced - [32] when auctioneer hosts and runs auction without deficit, individual rational — when a provider never gets a value less than the bid value (except when the provider cheats) [32], robust — when a compensation is given to the customer if the provider fails to deliver the promised QoS [33], optimal with respect to bidders, non-dominant — quotation proposed by a winner should be non-dominant with respect to all the providers those who are not the winners [34], incentive compatible — if truthful bidding is dominant strategy [12], Egalitarian socially welfare — which promotes those providers who lose in the auction to avoid the bidders drop problem and avoid control by some powerful providers. Further, FICA satisfies mobile data source and IOT needs.

4. Simulation study

The behavior of the proposed FICAM is evaluated by its simulation in various scenarios. As obvious, developing a real environment is not only a costly affair but difficult also due to the presence of several external entities such as service provider, broker, consumer, etc. Thus, simulation is a better choice. Fog computing in itself introduces challenges such as FICA, dynamic pricing model, location-wise Fog services, etc., therefore, existing simulators for Cloud computing environment i.e., CloudAuction [37], Cloudsim [38] cannot be used for the proposed work. Even the recently developed FogSim simulator does not include auctioning related simulation features. Therefore, simulation experiments have been designed in Matlab following the baseline model [10]. According to [10], none of the exiting simulators considers the requirements of the multi-attribute reverse auction for the emerging FICA. Also, according to [10], TFMCRA is the only truthful resource allocation model in Cloud computing with a multi-attribute combinatorial reverse auction mechanism, therefore, it has been considered for comparison purposes. The FICAM Matlab simulation procedure is presented in Appendix F.

4.1. Dataset

In order to simulate the proposed model and to analyze its behavior under the various scenario, the dataset is constructed from various publicly available sources [39]. However, to generate the dataset, standard pricing schemes of authenticated service providers are followed [39]. The dataset contains those providers who offer the services of both Fog and Cloud resources or those Fog providers who have tied up with Cloud provider and vice-versa. However, only Fog and only Cloud providers can participate as long as they can provide all types of VMs as required by the customer.

Thirty different types of VMs are considered [39], the configuration for which is briefed in Table 3. It is assumed that very high configuration VMs (index 15 to 30 in Table 3) are provided only on Cloud whereas small configuration VMs can be provided on Cloud or Fog by different providers. In the dataset, different types of VMs are indexed from 1 to 30 in increasing order of pricing based on their configuration. For example, 1 refers to VM with 512 MB RAM, 1 GB Storage, 1x CPU Power. We used $/minute price model of VMs as suggested in [40]. Dataset is constructed for 400 providers. The Cloud price of each provider’s VM Id 1 is randomly taken between [0.00748–0.0150]. Cloud price of other 29 VMs say VM Id v_x is randomly taken between [1.1125–1.2375] × (v_{x-1}). These VM Ids and their Cloud prices are generated corresponding to various service providers and their standard pricing rates from publicly available sources [39]. Technically, because of the limited number of Fog resources and high resource demand [41], it is observed that Fog prices are dynamic based on the customer requirement and load at the Fog provider’s site at any instance of time. In experiments, the fluctuation in Fog and Cloud prices are considered from −15% to 15%. Limit of mobility is local, state, and national just for the convention. 33% difference is taken between local Fog prices & state price, and 25% difference is taken between state & national Fog prices.

Table 3
VM types and configuration attributes.

VM ID	RAM	Disk	CPU	VM ID	RAM	Disk	CPU
1	512 MB	1 GB	1	16	4 GB	100 GB	5
2	1 GB	5 GB	1	17	4 GB	200 GB	6
3	1.5 GB	10 GB	1	18	8 GB	50 GB	6
4	2 GB	20 GB	1	19	8 GB	1 TB	5
5	512 MB	5 GB	2	20	4 GB	500 GB	6
6	1 GB	10 GB	2	21	2 GB	500 GB	5
7	2 GB	50 GB	2	22	16 GB	200 GB	4
8	1 GB	20 GB	3	23	4 GB	1 TB	4
9	1.5 GB	50 GB	3	24	16 GB	200 GB	8
10	4 GB	50 GB	2	25	8 GB	500 GB	6
11	4 GB	10 GB	3	26	32 GB	500 GB	6
12	2 GB	100 GB	3	27	16 GB	1 TB	6
13	2 GB	200 GB	3	28	32 GB	1 TB	7
14	8 GB	200 GB	1	29	32 GB	500 GB	8
15	8 GB	100 GB	4	30	64 GB	1 TB	8

Table 4
Common simulation parameters.

Types of VM(VM Id's)	1,3,4,6,7,9,12,15,17,19,22,23,25,28,30
Quantity of VM type	5,6,7,4,3,10,11,12,13,50,60,70,30,35,40
Time for VM type	12,14,16,17,18,19,20,40,60,120,90,50,100,70,80
QoS	1.0
Weight vector	0.25,0.25,0.25,0.25
Response time vector	0.3,0.35,0.4,0.5,0.6,0.7,0.9,1,1.5,2,2.5,3,3.5,4,5
Iterations	600
lm	local,state,national
Local probability	0.5
Remote probability	0.5
Reputation factor	2
Decrease factor	10

4.2. TFMCRA as baseline model

The resource allocation in the baseline model TFMCRA [10] is as follows. TFMCRA sorts providers based on price and non-price attributes. Sorted provider list is traversed until the constraint (in Eq. (9)) is not satisfied. All traversed providers and the Winners(x) (Winners(x) is p_x where constraint in Eq. (9) is satisfied) are declared as winners. Winner providers' resources are allocated to the customer.

$$\sum_{x=1}^{|Winners|} \sum_{i=1}^{n} q^i{}_{\text{Winners(x)}} >= q_c \tag{9}$$

Pricing in TFMCRA is as follows. In TFMCRA, providers bid bundle price. Let, $\mathrm{p_x}$ is a winner in TFMCRA and has bid $\mathrm{price_x}$. TFMCRA uses the Vickrey auction pricing scheme. After applying Vickrey pricing scheme, $\mathrm{price_x}$ becomes the price of runner-up provider i.e., p'_x. The Runner-up provider is the winner if p_x does not participate in the auction. The final price vector is calculated when the customer submits its QoS after using the resources. Broker imposes a penalty on the provider if the provider cheats a customer.

4.3. Results and discussion

The proposed FICAM model is evaluated for eight different scenarios. The experiments run 600 times; 200 times with lm = 'local', 200 times with 'state' and 200 times with 'national'. Results of these 600 iterations are averaged so that the effect of lm and non-price attributes can be conspicuously observed. In order to compare the results with the baseline model, the same simulation parameters i.e., quantity, type, prices, etc. as in the baseline model are used. These common simulation parameters (default) are listed in Table 4. The configuration of used VM types is listed in Table 3. Scenario 1 to 5 are listed here, whereas scenario 6 to 8 which study the effect of imposition of penalty on cheat providers, effect of priority attribute on providers, and effect of inclusion of reputation on provider is given in Appendices C, D, and E respectively.

4.3.1. Scenario1 : Resource procurement cost vs. discount

The customer receives some discount when it acquires a bundle of resources. The effect of variation of discount on resource procurement cost of FICAM and TFMCRA is analyzed. In this, the discount is varied from 0% to 50% on 50 providers. Other simulation parameters are as given in Table 4.

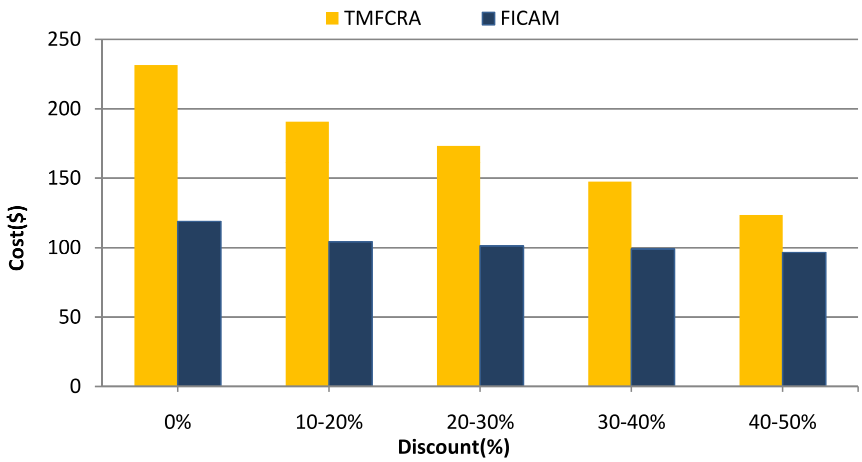


Fig. 3. A comparative study on resource procurement cost of FICAM & TFMCRA on varied discount.

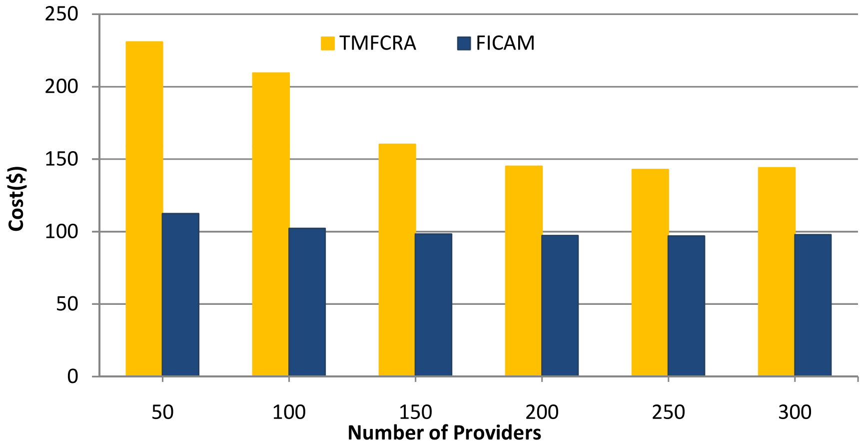


Fig. 4. Comparative cost of resource procurement cost of FICAM & TFMCRA on varied number of providers.

The following observations are derived from Fig. 3.

I. The resource procurement cost is less in FICAM than TFMCRA. It is because of the constraint, FICAM satisfies in Eq. (10), that considers availing only that much quantity of resources as required and not the whole bundle. TFMCRA satisfies constraint in Eq. (9) which allocated the quantity of the resources equal to or more than required by the customer.

$$\sum_{x=1}^{|Winners|} \sum_{i=1}^{n} q^{i}{}_{\mathrm{Winners(x)}} = q_c \tag{10}$$

II. As discount increases, resource procurement cost in TFMCRA decreases at a rapid rate than FICAM. The reason being FICAM allows customers to enjoy the discount only if one avails at least 50% of the resources offered in the provider's quote. Additionally, the discount is in the ratio of $\frac{qt}{qo}$ due to which the discount ratio decreases. Hence the number of winning providers, offering a discount to the customer, also decreases.

4.3.2. *Scenario 2: Resource procurement cost vs. number of providers*

In this experiment, resource procurement cost is observed on the varying number of providers. Similar to scenario 1, the algorithm is executed 600 times, and the results are averaged. Simulation parameters, except VM IDs [2, 3, 5, 7, 9, 11, 13, 15, 18, 20, 21, 23, 26, 28, 29], are same so that each VM type in the dataset can be utilized. From the results, shown in Fig. 4, it is observed that if the number of providers is increased, it will affect the resource procurement cost of both FICAM and TFMCRA models. However, the cost in FICAM is better in comparison to TFMCRA. It is also observed that the resource procurement cost decreases on the increase in the number of providers but it becomes constant after a certain number of providers. The reason for the same is that as the number of providers increases, the competition among the providers also increases resulting in a decrease in the cost. Resource procurement cost of FICAM is always less than TFMCRA because of the same reason as given in experiment 1.

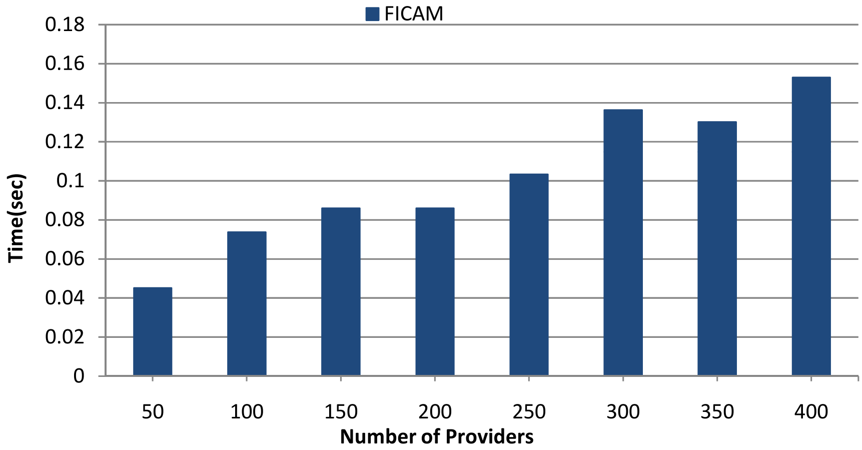


Fig. 5. Computation time of FICAM on varied number of providers.

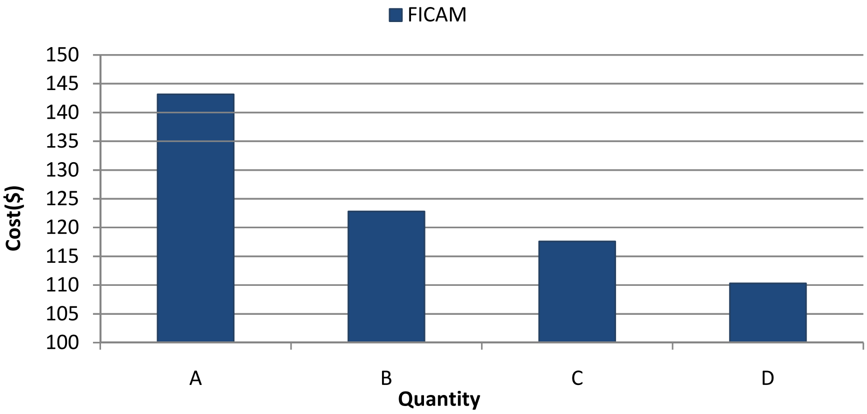


Fig. 6. Resource procurement cost of FICAM on varied quantity of resources.

4.3.3. Scenario 3: Computation time vs. number of providers

In scenario 3, the effect of a varied number of providers on computation time is observed. The simulation parameters, for this experiment, are same as in Table 4 except VM Ids $[1, 2, 4, 6, 7, 9, 12, 14, 16, 19, 22, 25, 27, 28, 30]$. Configuration of the machine to conduct this set of experiments is Intel core i5 with quad-core execution units, 4 MB DDR Cache and 4 GB RAM disk. It can be inferred from Fig. 5 that when the number of providers increases, computation time also increases which is obvious. It also shows that even when the number of providers is 400, computation time is less than one second. It can be concluded from the time complexity of FICAM (Appendix A), that the computation time of the FICAM is polynomial which fits very well for the computation time requirements.

4.3.4. Scenario 4: Resource procurement cost and number of winners vs. quantity

This experiment observes the behavior of the FICAM model in terms of resource procurement cost and the number of winners on varying quantities of the resource offered by the provider. The quantity of the resources, offered by the provider, is increased by an amount, and its effect is taken into account. Simulation parameters are same as in Table 4 expect VM Ids as $[2, 5, 8, 10, 11, 13, 14, 16, 18, 20, 21, 24, 26, 27, 29]$. In this scenario, 50 providers are accounted for bidding. Since Fog resources are limited in quantity [5], therefore, different quantity increment is considered for Fog and Cloud resources. In the first case, every provider is providing every Fog resource in the range of 0 to 5, and every Cloud resource in the range 0 to 10. In the second case, the quantity of every Fog resource is between 5 and 10, and every Cloud resource is between 10 and 20, i.e., an increment of 5 is considered for the Fog resources and 10 for the Cloud resources. Results are shown in Figs. 6 and 7. On the X-axis, 4 points are shown as, A, B, C, D, where point A corresponds to the first case i.e., where Fog resource range is [0–5] and Cloud resource range is [0–10], B corresponds to the second case and so on. From the results in Figs. 6 and 7, it can be observed that when the quantity of each VM type by the provider is increased, FICAM's resource procurement cost and the average number of providers decrease. The reason for the same is that the lowered price providers are now offering more resources at the same rate as before. So, FICAM acquires fewer resources from high-cost providers resulting in the decrease of cost and the number of winners.

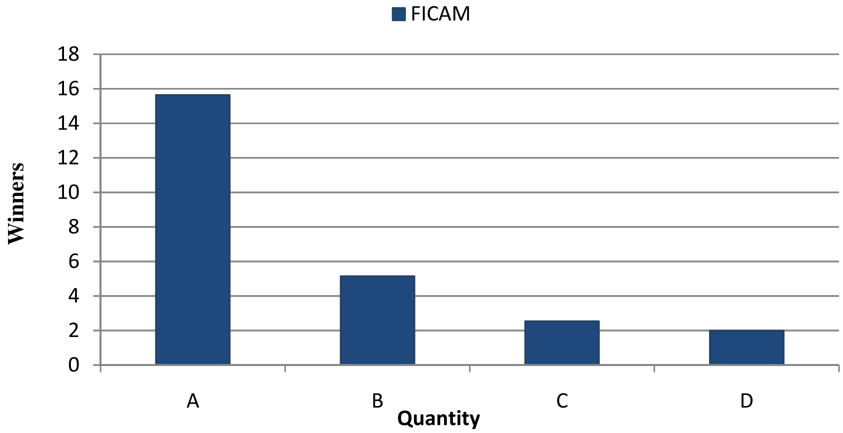


Fig. 7. Number of winners of FICAM on varied quantity of resources.

4.3.5. Scenario 5: Effect of limit_Mobility, probability, dynamic fog prices on resource procurement cost

(a) **Resource procurement cost vs. Limit_Mobility**: In this, FICAM behavior is observed on the limit of data source mobility. As discussed, the mobility limits may be local, state, or national. When mobility limit changes, remote prices of VM changes affecting the prices of Fog which also changes. To visualize this, when the mobility limit changes, the share of Fog price in the total amount is considerable. The share of Fog price in total amount, according to common simulation parameters, is less. Therefore, testing parameters are same as in Table 4 except quantity vector $[25, 30, 24, 27, 21, 26, 23, 32, 35, 37, 41, 39, 42, 38, 33]$. The expected time for which each VM acquired is 10 days. To understand the dynamism in the cost, the experiments are run two times with the same parameters, each with 600 rounds. As usual, out of 600, 200 rounds are for the local limit, 200 rounds for the state limit, and 200 rounds for the national limit. The resource procurement cost with local, state, and national mobility are shown in Fig. 8(a), where the local cost is the average cost of 200 rounds with lm = 'local'. State cost and National cost are similarly defined. From the results of these two 600 rounds, it comes out that the working behavior of the model is dynamic and the cost is non-deterministic.

(b) **Resource procurement cost vs. remote probability**: This scenario estimates the effect of change of data source remote probability on resource procurement cost. Simulation parameters are the same as in scenario 5(a), and each time algorithm runs for 600 rounds — 200 for local, 200 for the state, and 200 for national. The average cost of 600 rounds is taken into account. Remote probability is varied from 0 to 1 to observe the behavior of FICAM. The results are shown in Fig. 8(b).

(c) **Effect of dynamic fog price on resource procurement cost**: As load changes on Fog resources, their prices are varied (increase/decrease) due to dynamic pricing [41]. This experiment estimates the effect of dynamic Fog pricing on total resource procurement cost. For this set of experiments, the percentage difference between local Fog price and Cloud price of each VMs of each provider is varied from −30% to 50%. Other parameters are the same as in scenario 5(a). Each experiment is conducted 600 times as in scenario 5(b). The effect of the cost of dynamic Fog pricing is shown in Fig. 8(c).

From the results in Fig. 8(a), (b) and (c), it is inferred that the working behavior of the model is dynamic. Providers in FICAM are sorted on the basis of total estimated bundle cost as given in Eq. (1). Whenever lm(as per customer mobility) or remote probability or Fog prices increase, tlp and trp are multiplied by some factor but, tcp remains constant. Total estimated bundle price increases which changes the sorted list. From Theorem 1, it can be proved that the place of provider in the new sorted list may not be the same as in previously sorted list i.e., list before lm or remote probability or Fog price changes. This changes the winner which affects the resource procurement cost. Resource procurement cost can increase or decrease, i.e., no direct relation can be established between resource procurement cost and the mobility limit or the remote probability or dynamic Fog prices. An example can demonstrate this better. Suppose, there are six providers i.e., a, b, c, d, e, and f. The sorted list of providers before the increase is [a, b, c, d, e, f] and winners are [a, b, c, d]. After the increase, the sorted list of providers is [a, b, c, e, f, d] and winners are [a, b, c, e]. Total resource procurement cost before the increase is given by Eq. (11).

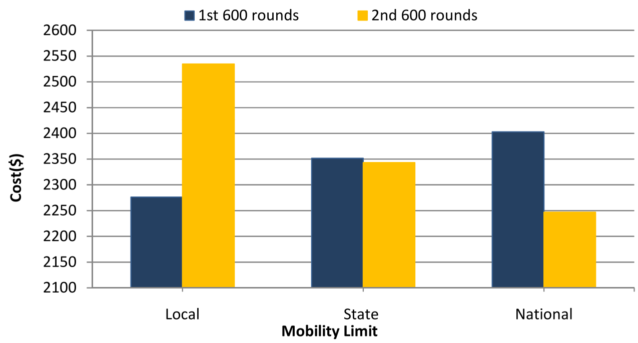
$$fb_{\mathrm{a}} + fb_{\mathrm{b}} + fb_{\mathrm{c}} + fb_{\mathrm{d}} \tag{11}$$

After increase, the resource procurement cost is as given in Eq. (12)

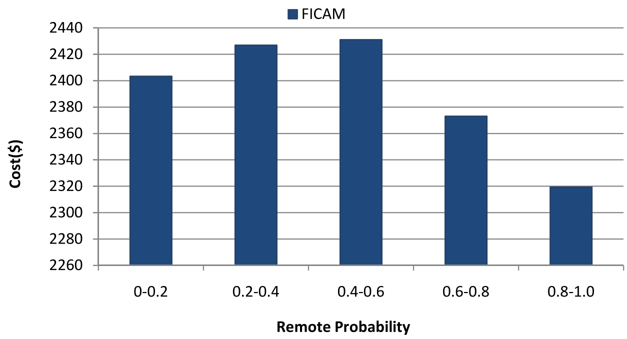
$$fb'_{\mathrm{a}} + fb'_{\mathrm{b}} + fb'_{\mathrm{c}} + fb'_{\mathrm{e}} \tag{12}$$

The final bill of provider a after the increase may be greater than or equal to before the increase (equal to in case when the provider is offering Cloud resources only) i.e., $fb'_{\mathrm{a}} >= fb_{\mathrm{a}}$. This is valid for providers b *and* c as well. Thus, it is possible to say

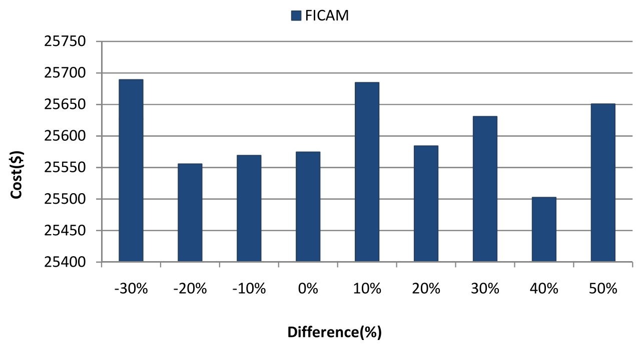
$$fb_{\mathrm{a}} + fb_{\mathrm{b}} + fb_{\mathrm{c}} <= fb'_{\mathrm{a}} + fb'_{\mathrm{b}} + fb'_{\mathrm{c}} \tag{13}$$



(a) Cost vs Limit_Mobility



(b) Cost vs Remote Probability



(c) Cost vs Dynamic Fog Price

Fig. 8. Resource procurement cost of FICAM on varied Limit_Mobility, probability, dynamic fog price.

However, it is difficult to say $fb'_{\mathrm{e}} >= fb_{\mathrm{d}}$ for providers d and e because providers are sorted not only on price but also on the basis of non-price attributes. fb'_{e} can be less than fb_{d}. Hence, total resource procurement cost before the increase, can be less than total resource procurement after the increase, i.e., Eq. (14) or Eq. (15).

$$fb_{\mathrm{a}} + fb_{\mathrm{b}} + fb_{\mathrm{c}} + fb_{\mathrm{d}} >= fb'_{\mathrm{a}} + fb'_{\mathrm{b}} + fb'_{\mathrm{c}} + fb'_{\mathrm{e}} \tag{14}$$

$$fb_{\mathrm{a}} + fb_{\mathrm{b}} + fb_{\mathrm{c}} + fb_{\mathrm{d}} <= fb'_{\mathrm{a}} + fb'_{\mathrm{b}} + fb'_{\mathrm{c}} + fb'_{\mathrm{e}} \tag{15}$$

Theorem 1. *If tlp is increased by the factor a, and trp is increased by factor b then place of providers in new sorted list changes.*

Proof. Let, $p_x \& p_y$ be two providers, and before price increases, place of p_x appears before p_y in the sorted list of providers. Therefore,

$$tlp_{\mathrm{x}} + trp_{\mathrm{x}} + tcp_{\mathrm{x}} < tlp_{\mathrm{y}} + trp_{\mathrm{y}} + tcp_{\mathrm{y}} \tag{16}$$

When tlp increases by the factor d1, and trp increases by a factor d2 then sorted list changes. It is undetermined that place of provider p_x will remain before p_y in the new sorted list of providers because Eq. (17) cannot be derived from Eq. (16).

$$tlp_{\mathrm{x}} \times d1 + trp_{\mathrm{x}} \times d2 + tcp_{\mathrm{x}} < tlp_{\mathrm{y}} \times d1 + trp_{\mathrm{y}} \times d2 + tcp_{\mathrm{y}} \tag{17}$$

Hence, the place of provider in the new sorted list changes.

5. Conclusion and future work

In this paper, a novel multi-attribute auctioning mechanism for resource procurement considering fog-integrated cloud architecture is proposed. The combinatorial reverse auction is applied for resource procurement as it is more beneficial than its counterparts. Considering Fog and Cloud service providers and their respective attributes, an architecture model is proposed along with pricing design and resource procurement algorithm. Pricing design is based on local, remote Fog resources, and Cloud resources. The proposed FICAM model takes care of data source mobility and issues of Fog resource limitations. It estimates the bundle price of each provider on the basis of the customer's local and remote probability, the limit of mobility, and determines the winners according to the estimated bundle price and other attributes. Unlike the Cloud, Fog resources are limited and therefore, the algorithm determines and takes into account the number of resources to be acquired from each winner provider. An incentive scheme for the winner providers is also proposed. The final bill is generated based on local, remote, Cloud time matrices, and QoS delivered. For simulation purposes, a dataset is prepared because of the unavailability of the required dataset that consists of Fog and Cloud reverse auction prices considering the data source mobility in Fog computing. Eight different scenarios are explored to observe the behavior of FICAM considering various consequences. Results are compared with the baseline Cloud auction model. The experimental results infer that FICAM works effectively in real-time and in a highly dynamic scenario. Thus, it fulfills all the features to be a good candidate as the future auctioning model that meets the requirements of customers and service providers.

In the future, we plan to apply machine learning techniques to make the FICAM a more truthful, realistic, memorable auctioning mechanism that can predict the behavior of the service provider based on the experiences.

Acknowledgment

We acknowledge and thank Dr. Gaurav Baranwal for his kind support in this work.

Appendix A. Time complexity analysis

Let, m is the total number of providers participating in the auction, and n is the number of different VM configurations that a customer requires. FICAM collects the requirement from the customer (line 1) with the time complexity of $O(n)$. Next, it seeks the quotation from the providers with the complexity of $O(m \times n)$ (lines 3–5) that is extended by including the time-complexity of $O(m)$. Further, the *'Estimated Bundle Price Calculator'* module (line 9) has $O(m \times n)$ complexity. The '*Winner & Quantity Determination*' module is called (line 10) to accomplish two jobs: sorting of providers and winner estimations are done in $O(m \times n)$ and $O(m^2)$ time-complexity respectively. The incentive is given to truthful winners through *'Incentivize Winning Providers'* module (line 11) which calculates runner-up winner and also calls *'Winner & Quantity Determination'* module (line 5) for each winning provider. Therefore, total complexity becomes $O(m^2 + m^2 \times n + m^3)$. FICAM asks provider to submit time taken matrices (lines 13–15) which have the $O(m \times n)$ time-complexity. Finally, the bill is generated through *'Price Settlement'* module (line 16) with the time-complexity of $O(m \times n)$. Hence, FICAM is running in polynomial time with estimated time complexity: $\mathrm{O}(\mathrm{m}^2\mathrm{n} + \mathrm{m}^3)$. Normally, m is very large compared to n, therefore, total-time complexity becomes $O(m^3)$.

Appendix B. Proof of theorems

Theorem 2 (*FICAM is Truthful or Incentive Compatible*)**.** *A model is said to be incentive-compatible if truthful bidding is the dominant strategy [12]. The payoff of provider/bidder is given by Eq. (B.1) [12] if provider i is a winner otherwise his/her payoff is zero.*

$$payoff_i = fp_i - v_i \tag{B.1}$$

Here, fp is the final price, and v is the valuation price. In order to establish the truthfulness of FICAM, it is required that if the provider overbids or underbids than his valuation value (truthful value), the provider should be in loss or zero profit than if the provider had bid truthfully. Let us assume, b_i be the bid value of provider i and v_i be the valuation value of provider$_i$.

Table B.5
Common simulation parameters.

Types of VM(VM Id's)	1,3,4,6,7,9,12,15,17,19,22,23,25,28,30
Quantity of VM type	5,6,7,4,3,10,11,12,13,50,60,70,30,35,40
Time for VM type	12,14,16,17,18,19,20,40,60,120,90,50,100,70,80
QoS	1.0
Weight vector	0.25,0.25,0.25,0.25
Response time vector	0.3,0.35,0.4,0.5,0.6,0.7,0.9,1,1.5,2,2.5,3,3.5,4,5
Iterations	600
lm	local,state,national
Local probability	0.5
Remote probability	0.5
Reputation factor	2
Decrease factor	10

Case 1: If provider$_i$ can truthfully win the auction with $b_i = v_i$, runner-up as r_i and payoff as $payoff_i$.

(a) If provider$_i$ overbids, i.e., $b_i > v_i$ then he may lose or win; if the provider loses, i.e., $payoff_i = 0$ so, there is no profit of overbidding. Else, if provider wins then, runner-up of provider$_i$ will still be r_i as technically runner-up winner of provider$_i$ is the winner if provider$_i$ did not participate. Hence, provider$_i$ payoff will be $payoff_i$.

(b) If the provider underbids, i.e., $b_i < v_i$ then, he/she will win. But runner-up winner and payoff will even now be r_i and $payoff_i$ respectively. Hence, no profit of underbidding.

Case 2: If provider$_i$ with $b_i = v_i$ cannot win the auction with Winner Providers (WPs).

(a) If provider$_i$ underbids, i.e., $b_i < v_i$ then, he can lose or win; if the provider loses, i.e., $payoff_i = 0$ so, no profit of underbidding. Else if provider wins then, the runner-up winner of provider$_i$ will be among WP. Since, provider i is non-winner with $b_i = v_i$, therefore, $v_i > max(bid(WP))$. Following this, $payoff_i$ is < 0. Hence, loss is there if provider underbids.

(b) If the provider overbids, i.e., $b_i > v_i$, he will lose with $payoff_i = 0$. Hence, there is no profit of overbidding.

With this, we can infer, truthfulness dominates underbidding and overbidding, therefore, FICAM is truthful.

Theorem 3 (*FICAM is Monotone*). *A model is a monotone if a provider wins even if the provider improves its quotation. The provider's quotation consists of initial price vectors, quantity offered vector, discount on the bundle, and QoS. The proof is given for all possible cases, if the provider improves any part of the quotation (improvement is done with respect to the customer) then the provider wins.*
(a) lp, rp and cp consist of initial local, remote and Cloud price vector. Improvement in any of these vectors leads to decrement in the estimated cost of the bundle; in turn, it decreases the Bid Density (Eq. (11)) and point (Eq. (13)). Since, provider's list is sorted on the basis of the point, the decrement of the point ensures that the provider still wins.
(b) If qo is improved (the provider offers more quantity) then M(x) (Eq. (10)) decreases. This results in a decrease in Bid Density (Eq. (11)) that leads to a decrement in point Eq. (13) which ensures the victory of the provider.
(c) If $discob$ is improved (discount is increased) then the estimated bundle price decreases leading to a decrease in point of the provider. Hence, the provider still wins.
(d) QoS is improved. It improves score (Eq. (12)) which leads to an improvement in point as in Eq. (13).

Theorem 4 (*FICAM is Non-dominant*). *It means that the quotation proposed by a winner should be non-dominant with respect to all non-winner providers. Nevertheless, quotation needs not to be non-dominant with respect to winner providers [34]. Non-dominant quotation means quotation should be best according to sorting/comparison criteria of the provider. Since the winner determination algorithm starts from the first provider in the sorted list (the best provider according to sorting criteria) to declare winners and continues till it achieves the desired quantity of resources. All providers declared as winners, have the best quotation compared to non-winner providers. This indicates FICAM is non-dominant.*

Theorem 5 (*FICAM is Robust*). *A combinatorial auction is robust if it compensates the customer for non-meeting the promised QoS as mentioned in the quotation [33]. To accomplish this, the score is recalculated to determine whether the provider has offered the stated QoS. If the provider fails then a penalty is imposed and the customer pays according to the delivered QoS.*

Theorem 6 (*FICAM Ensures Egalitarian Social Welfare*). *It ensures that the model should promote losing providers, i.e., should increase chances of winning of non-winner providers in the next auction round to avoid control by some powerful providers in the market. In FICAM, the priority attribute takes care of this, therefore, FICAM ensures egalitarian social welfare.*

Theorem 7 (*FICAM is Budget Balanced [32]*). *When auctioneer hosts and runs auction without deficit, the auction is said to be budget balanced. In the 'Price Settlement' module of FICAM, the final amount received from the customer is paid to the provider. Therefore, FICAM is budget balanced.*

Theorem 8 (*FICAM is Individually Rational*). *A model is individually rational [32] when a provider never gets a value less than the bid value (except when the provider cheats). FICAM provides either incentive > 0 or equal to 0 if the provider's runner-up winner does not exist. This way, the final price after the incentive can never be less than the initial bid price. Hence, FICAM is individually rational.*

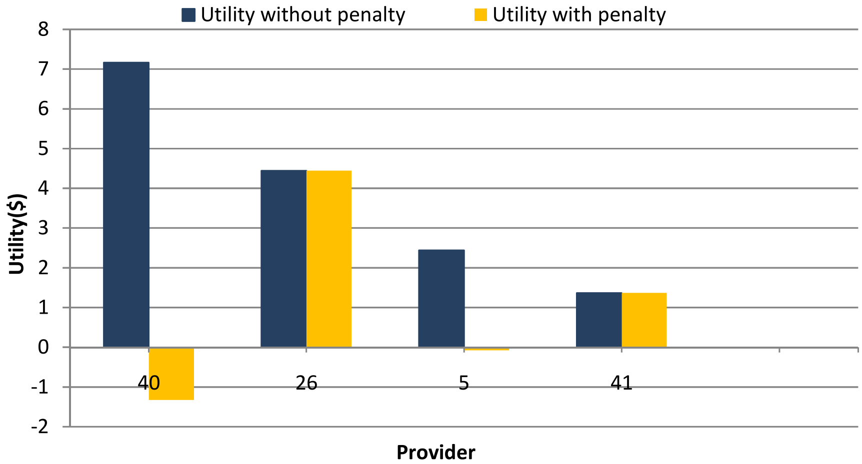


Fig. C.9. Utility of providers on the imposition and not the imposition of penalty on fraud providers.

Table D.6
Effect of reputation on providers.

Round	Winners
1	[5, 4, 3, 10]
2	[5, 4, 3, 7]
3	[5, 4, 3, 7]
4	[5, 4, 3, 7]
5	[5, 4, 3, 10]

Appendix C. Effect of imposition of penalty on cheating providers

Fig. C.9 shows the effect of the penalty on providers. The simulation parameters are same as in Table B.5. From Fig. C.9, it can be observed that the provider number 40 and 5 have negative utility and 26 and 41 have positive utility when the penalty is applied. The utility of provider is given by Eq. (C.1) where fap is the final auction price, and bp is the bid price.

$$\text{Utility(x)} = fap(x) - bp(x) \tag{C.1}$$

Let us assume, providers 40 and 5 cheat; therefore, no incentive is given. In the price settlement module, the final score of cheating provider is not equal to the initial score. As a penalty, their final auction price will be according to their bid prices, and it is multiplied by ratio $\frac{final\_score}{initial\_score}$ which is less than 1. This way, the final auction price becomes less than the price according to bid prices which indicates negative utility. From the results in Fig. C.9, it can also be observed that if the penalty is not imposed on cheating (fraud) providers (40 and 5), their utility seems to be positive. In addition, if the penalty is not imposed on cheating, then provider again commits fraud in the next auction rounds. Providers 26 and 41 have positive utility because their final scores are equal to the initial score. Consequently, they receive the incentive, and hence, the final price is higher than the bid price. As an inference, penalty imposition discourages provider from perpetrating fraud.

Appendix D. Effect of priority attribute on provider

In this scenario, the effect of the priority attribute on provider is analyzed. Simulation parameters are the same as in Table B.5 except that the effect of reputation is not taken into account to consider the effect of priority attribute. A total of 10 providers is considered here. The experiment is run 200 times. The number of times a provider wins in these 200 rounds is recorded and is shown in Fig. D.10. The sorted list of providers based on their quotation is $[5, 4, 3, 10, 7, 9, 1, 8, 2, 6]$. The first-round winner providers are $[5, 4, 3, 10]$. However, it can be inferred from Fig. D.10 that providers 7 and 9 are also able to win a substantial number of the auction even though they are not offering the best bids. Even, providers 1 and 8 are able to win in some auction rounds. This happened because of the inclusion of priority. Therefore, not only most competent providers always win, but a fair chance is also given to other providers in FICAM. Hence, the priority attribute helps to establish a balance in the market.

Appendix E. Effect of inclusion of reputation on providers

The effect of reputation attributes on providers is also analyzed. Simulation parameters are the same as in Table B.5 except that effect of priority is disabled to analyze the effect of reputation attribute. The experiment is performed with 10 providers, and the df(decreasing factor) is 50. Results from Table D.6 shows that in round 1, providers $[5, 4, 3, 10]$ win. Let us consider,

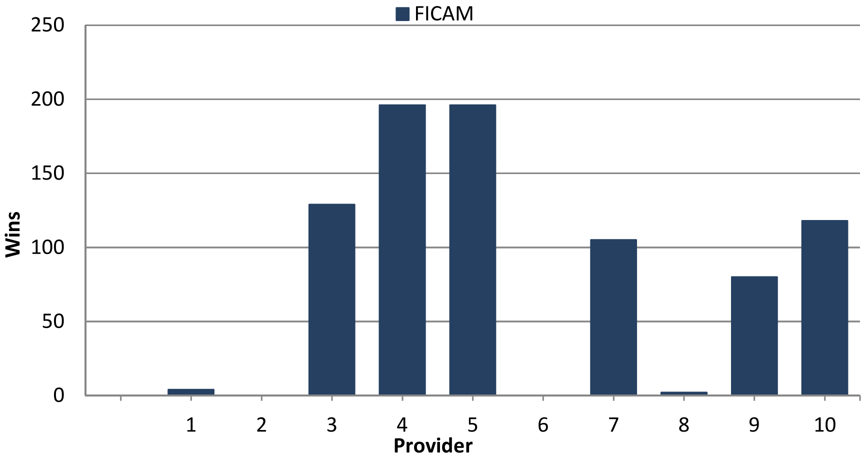


Fig. D.10. Effect of priority attribute on providers.

providers 4 and 10 gave false QoS; therefore, their reputation decrease. Providers 4 and 10 promised a QoS value of 1 and assumed, they delivered 0.9613 and 0.9363 QoS respectively. In the next round, results showed that providers $[5, 3, 7, 9]$ win because of the decreased reputation of providers 4 and 10. Logically, if providers 4 and 10 decrease their quotation price or increase their QoS truthfully then, they can win in the next auction rounds. The same reputation effect, using FICAM, is also observed. To analyze this behavior of FICAM, the quotation price of providers 4 and 10 are decreased by 2%, and the QoS is increased by 10%. This rate of improvement is continued until a provider is again able to win in the auction. The set of experiments is performed until the winner's list becomes the same as the first round's winner list. The obtained experimental results are shown in Table D.6. From the results, it can be inferred that the provider improved its quotation and price in the second round; therefore, provider number 4 was able to win in the second round. Provider number 4 was truthful, therefore, provider 4 was able to win in the next auction round too. Provider 10 was more untruthful (with respect to delivered QoS), despite reducing quotation price and increasing quality in every round, provider number 10 was able to win again (5th round). With this study, it can be inferred that the decrease in reputation attributes can influence the providers a lot.

Appendix F. Matlab simulation procedure

To simulate FICAM in Matlab, the dataset for 400 providers is generated. For each provider p_x, a random number between 0 and 15 is generated. p_x will provide VM Ids up to this random number on Fog and after this number, on the Cloud. This random number is generated so that different providers offer a different number of VM Ids on Fog and Cloud. For generating final local, remote, Cloud time matrices, the following pseudo-code is used.

if $lm = local$ **then**
  for each VM Id i
    if i is on Fog **then**
      // final remote time and Cloud time will be zero
      $flt^{i}_{x} = et^{i}$
    else
      // final local time and remote time will be zero
      $fct^{i}_{x} = et^{i}$
    end
  end
else
  for each VM Id i
    if i is on Fog **then**
      // final Cloud time will be zero
      $flt^{i}_{x} = et^{i} * p_l$
      $frt^{i}_{x} = et^{i} * p_r$
    else
      // final local time and remote time will be zero
      $fct^{i}_{x} = et^{i}$
    end
  end
end

where p_x is any random winner provider.

Appendix G. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.simpat.2021.102307.

References

[1] A. Dastjerdi, R. Buyya, Fog Computing: Helping the Internet of Things Realize its Potential, IEEE Computer,IEEE CS Press, USA, 2016, pp. 40–44.
[2] S. Yi, Z. Hao, Z. Qin, Q. Li, Fog computing: Platform and applications, in: Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), 2015, pp. 73–78, http://dx.doi.org/10.1109/HotWeb.2015.22.
[3] L. Yu, T. Jiang, Y. Zou, Fog-assisted operational cost reduction for cloud data centers, IEEE Access (2017) 13578–13586.
[4] A. Markus, A. Kertesz, A survey and taxonomy of simulation environments modelling fog computing, Simul. Model. Pract. Theory 101 (2020) 102042.
[5] Fog computing and the internet of things: Extend the cloud to where the things are, White paper C11-734435-00, 2015, pp. 1–6, https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf.
[6] K. Al-Zoubia, G. Wainer, Fog and cloud collaboration to perform virtual simulation experiments, Simul. Model. Pract. Theory 101 (2020) 102032.
[7] G. Manoochehri, C. Lindsy, Reverse auction: Benefits, challenges, best practices, Calif. J. Oper. Manage. (2008) 123–130.
[8] N. Kumar, D. Vidyarthi, An energy aware cost effective scheduling framework for heterogeneous cluster system, Future Gener. Comput. Syst. (2017) 73–88.
[9] T. Schoenherr, V.A. Mabert, Online reverse auction: common myths vs evolving reality, Bus. Horiz. (2007) 373–384.
[10] G. Baranwal, D. Vidyarthi, A truthful and fair multi-attribute combinatorial reverse auction for resource procurement in cloud computing, IEEE Trans. Serv. Comput. (2016) 77–97, http://dx.doi.org/10.1109/TSC.2016.2632719.
[11] Y. Abuseta, A fog computing based architecture for IoT services and applications development, Int. J. Comput. Trends Technol. 67 (2019) 92–98.
[12] https://en.wikipedia.org/wiki/Vickrey_auction.
[13] N. Nisan, A. Ronen, Algorithmic mechanism design, in: Proceeding STOC '99 Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, 1999, pp. 129–140.
[14] T.H. Luan, L. Gao, Z. Li, Y. Xian, G. We, L. Sun, Fog computing: Focusing on mobile users at the edge, Eng. Comput. (2016) 1–11, doi:arXiv:1502.01815v3.
[15] S. Sarkar, S. Chatterjeei, Assessment of the suitability of fog computing in the context of internet of things, IEEE Trans. Cloud Comput. (2015) 46–59, http://dx.doi.org/10.1109/TCC.2015.2485206.
[16] Q. Fan, N. Ansari, Assessment of the suitability of fog computing in the context of internet of things, IEEE Trans. Netw. Sci. Eng. (2018) 1, http://dx.doi.org/10.1109/TNSE.2018.2852762.
[17] Y. Jiao, P. Wang, D. Niyato, K. Suankaewmanee, Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks, IEEE Trans. Parallel Distrib. Syst. (2019) 1, http://dx.doi.org/10.1109/TPDS.2019.2900238.
[18] M. Debe, K. Salah, Blockchain-based decentralized reverse bidding in fog computing, IEEE Access 8 (2020) 81686–81697.
[19] B. Song, M. Hassan, E. Huh, A novel cloud market infrastructure for trading service, in: Computational Science and Its Applications,2009,ICCSA'09, International Conference on, IEEE, 2009, pp. 44–50.
[20] S. Vries, R. Vohra, Combinatorial auctions: A survey, INFORMS J. Comput. (2003) 284–309.
[21] G. Prasad, S. Rao, A combinatorial auction mechanism for multiple resource procurement in cloud computing, in: Intelligent Syaytems Design and Applications(ISDA), 2012 12th Inter-national Conference on IEEE 2012, 2012, pp. 337–344.
[22] M. Liwang, S. Dai, A truthful reverse-auction mechanism for computation offloading in cloud-enabled vehicular network, IEEE Internet Things J. 6 (2019) 4214–4227.
[23] A. Pla, B. Lopez, J. Murillo., Multi-attribute auctions with different types of attributes: Enacting properties in multi-attribute auctions, Expert Syst. Appl. (2014) 4829–4843.
[24] J.S. Lee, B.K. Szymanski, A novel auction mechanism for selling time-sensitive e-services, E-commerce technology, in: Seventh IEEE International Conference on IEEE, 2005, pp. 75–82.
[25] L.F. Bittencourt, J. D., R. Buyya, O.F. Rana, M. Parashar, Mobility-aware application scheduling in fog computing, IEEE Cloud Comput. (2017) 26–35.
[26] R.K. Naha, S. Garg, D. Georgakopoulos, P.P. Jayaraman, L. Gao, Y. Xiang, R. Ranjan, Fog computing: Survey of trends, architectures, requirements, and research directions, IEEE Access 6 (2018) 47980–48009.
[27] Y.-L. Jiang, Y.-S. Chen, S.-W. Yang, C.-H. Wu, Energy-efficient task offloading for time-sensitive applications in fog computing, IEEE Syst. J. PP (2018) 1–12, http://dx.doi.org/10.1109/JSYST.2018.2877850.
[28] https://en.wikipedia.org/wiki/Roaming/.
[29] D. Lehmann, L. Ocallaghan, Y. Shoham, Truth revelation in approximately efficient combinatorial auctions, J. ACM (2002) 577–602.
[30] https://en.wikipedia.org/wiki/Combinatorial_auction.
[31] https://en.wikipedia.org/wiki/Multi-attribute_auction.
[32] A. Jin, W. Song, P. Wang, D. Niyato, P. Ju, Auction mechanisms toward efficient resource sharing for cloudlets in mobile cloud computing, in: In: IEEE Transactions on Services Computing, 2016, pp. 895–909.
[33] A. Holland, B. O'Sullivan, Robust solutions for combinatorial auctions, in: In: 6th ACM Conference on Elec-Tronic Commerce, ACM, 2005, pp. 183–192.
[34] M.J. Bellosta, S. Kornman, D. Vanderpooten, Preference-based english reverse auctions, Artificial Intelligence (2011) 1449–1467.
[35] A. Prasad, S. Rao, A mechanism design approach to resource procurement in cloud computing, Comput. IEEE Trans. (2014) 17–30.
[36] G. Modica, G. Petralia, O. Tomarchio, Procurement auctions to trade computing capacity in the cloud, in: P2P, Parallel, Grid, Cloud and Internet Computing(3PGCIC), 2013 Eighth International Conference on, IEEE, 2013, pp. 298–305.
[37] P. Samimi, Y. Teimouri, M. Mukhtar, A combinatorial double auction resource allocation model for cloud market, Inform. Sci. (2014).
[38] R. Calheiros, R. Ranjan, A. Beloglazov, C.D. Rose, R. Buyya, Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Softw. - Pract. Exp. (2011) 23–50.
[39] https://www.cloudorado.com/.
[40] http://omerio.com/2016/03/16/saving-hundreds-of-hours-with-google-compute-engine-per-minute-billing/.
[41] H. Zhang, Y. Xiao, S. Bu, D. Niyato, R. Yu, Z. Han, Computing resource allocation in three-tier IoT fog networks: a joint optimization approach combining stackelberg game and matching, IEEE Internet Things J. (2017) 1204–1215.

Anubha Aggarwal received B.Tech. degree in Computer Science & Engineering from Shri Mata Vaishno Devi University, Katra, J&K, India in 2017. Currently, she is working in TaDigital as Software Engineer. Her research interests include resource provisioning and management in Fog Computing, Cloud Computing, Question Answering Systems and Word Alignment.

Dr. Neetesh Kumar received his M.tech and Ph.D. degrees from the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi. He is currently a Faculty Member with the Indian Institute of Technology (IIT) Roorkee, India. He has published many research publications in the world's top tier publishers like IEEE journals and transactions (on GCN, ITS), Elsevier journals and Springer Journals, etc. He is a regular reviewer of highly reputed journals like IEEE Transactions on Parallel and Distributed Systems, IEEE Journal of Internet of things, FGCS, etc. He is also acting as a lead PI for several sponsored projects from DST/CSIR agencies, Government of India. He has been a technical program committee member in several conferences. Broadly, His research interests include algorithm design, Intelligent Transportation System (ITS), high-performance computing, soft computing, and computational intelligence, Cloud/Fog computing, IoT, Software Defined Networking.

Deo Prakash Vidyarthi is working as Professor in the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi. Dr. Vidyarthi has published around 95 research papers in various peer-reviewed Journals and Transactions (including IEEE, Elsevier, Springer, Wiley, World Scientific, etc.) and around 50 research papers in proceedings of peer-reviewed conferences. He has contributed chapters in many edited books. He is on the editorial board of many International Journals and also in the reviewer's panel of many International Journals. Dr. Vidyarthi has co-authored a book (research monograph) entitled "Scheduling in Distributed Computing Systems: Design, Analysis and Models" published by Springer, the USA released in 2009. Another book (edited) by Dr. Vidyarthi is "Technologies and Protocols for the Future Internet Design: Reinventing the Web", by IGI-Global (USA) released in the year 2012. The third book from Dr. Vidyarthi is "Auction based Resource Provisioning in Cloud Computing" by Springer was released in 2018. Dr. Vidyarthi is a senior member of the IEEE, ACM, International Society of Research in Science and Technology (ISRST), USA, International Association of Computer Science and Information Technology (IACSIT), Singapore, and International Association of Engineers (IAENG). His research interest includes Parallel and Distributed System, Grid and Cloud Computing, Mobile Computing, Evolutionary Computing, etc.

Rajkumar Buyya is a Redmond Barry Distinguished Professor and Director of the Cloud Computing and Distributed Systems (CloudS) Laboratory at the University of Melbourne, Australia. He served as a Future Fellow of the Australian Research Council during 2012–2016. He has authored over 625 publications and seven textbooks. He is one of the highly cited authors in computer science and software engineering worldwide. Dr. Buyya has led the establishment and development of key community activities, including serving as foundation Chair of the IEEE Technical Committee on Scalable Computing and five IEEE/ACM conferences. He served as the founding Editor-in-Chief of the IEEE Transactions on Cloud Computing. He is currently serving as Co-Editor-in-Chief of Journal of Software: Practice and Experience. Dr. Buyya is recognized as a "2016 Web of Science Highly Cited Researcher" by Thomson Reuters, a Scopus Researcher of the Year 2017 with Excellence in Innovative Research Award by Elsevier, and a Fellow of IEEE for his outstanding contributions to Cloud computing.