


RESEARCH

Open Access

A cloud priority-based dynamic online double auction mechanism (PB-DODAM)



S. M. Reza Dibaj^{1*†} , Ali Miri^{1†} and SeyedAkbar Mostafavi^{2†}

Abstract

Double auctions are considered to be effective price-scheduling mechanisms to resolve cloud resource allocation and service pricing problems. Most of the classical double auction models use price-based mechanisms in which determination of the winner is based on the prices offered by the agents in the market. In cloud ecosystems, the services offered by cloud service providers are inherently time-constrained and if they are not sold, the allocated resources for the unsold services are wasted. Furthermore, cloud service users have time constraints to complete their tasks, otherwise, they would not need to request these services. These features, perishability and time-criticality, have not received much attention in most classical double auction models. In this paper, we propose a cloud priority-based dynamic online double auction mechanism (PB-DODAM), which is aligned with the dynamic nature of cloud supply and demand and the agents' time constraints. In PB-DODAM, a heuristic algorithm which prioritizes the agents' asks and bids based on their overall condition and time constraints for resource allocation and price-scheduling mechanisms is proposed. The proposed mechanism drastically increases resource allocation and traders' profits in both low-risk and high-risk market conditions by raising the matching rate. Moreover, the proposed mechanism calculates the precise defer time to wait for any urgent or high-priority request without sacrificing the achieved performance in resource allocation and traders' profits. Based on experimental results in different scenarios, the proposed mechanism outperforms the classical price-based online double auctions in terms of resource allocation efficiency and traders' profits while fulfilling the double auction's truthfulness pillar.

Keywords: Priority-based dynamic online double auction, Mechanism design, Cloud secondary market, Dynamic resource allocation

Introduction

The cloud ecosystem is a business model that needs an appropriate pricing mechanism to satisfy service providers, as well as service users to survive and grow in the current competitive markets [1]. One of the most prevalent pricing methods is an on-demand pricing model. Since in the on-demand pricing model, customers of cloud computing resources have full control over their operational costs, and they can start and end the use of resources according to their needs, the on-demand pricing model is desirable for them [2, 3]. This mechanism

is not equally able to meet the interests of cloud service providers, as they are interested in planning and preparing for the future by buying the necessary resources and services in advance. An accurate estimation of the future needs cannot be achieved to service providers by the on-demand pricing method. The dramatic changes in cloud environments and the lack of a long history of using cloud services make it difficult for service providers to predict accurately and appropriately the capacity of needed resources and services. Furthermore, upgrading technological infrastructure is extremely costly for service providers. Since the capacity and quality of computing technologies are increasing every day, and their prices are constantly decreasing, service providers would be better off postponing infrastructure upgrade as much as

*Correspondence: smreza.dibaj@ryerson.ca

[†]S. M. Reza Dibaj, Ali Miri and SeyedAkbar Mostafavi contributed equally to this work.

¹Department of Computer Science, Ryerson University, Toronto, Canada
Full list of author information is available at the end of the article

possible [4]. Predicting the right time to upgrade their infrastructure is a difficult task that the on-demand pricing model further complicates. Moreover, the appropriate number of support staff [5] and the required power [6] are among the variable parameters that are achieved only by knowing the number of users' service requests in advance. On the one hand, over-investing in infrastructure and support staff is a waste of resources. On the other hand, not providing the users' needed resources and support staff cause Service Level Agreement penalties for the providers and damage their reputations. Among the cloud services, scheduling Virtual Machine (VM) instances efficiently in IaaS is a challenging problem if users do not indicate their requests in advance [7, 8]. Calculating the amount needed to invest in infrastructure and the right number of support staff, and scheduling the VMs efficiently, require sufficient knowledge of the needs of users in future time windows, which on-demand pricing models cannot provide.

Futures contracts and options contracts are two alternative pricing models to ensure that users have access to their needed resources in the future, usually by paying a reduced price in advance. The former method obliges users to own what they have bought, while in the latter method users have the legal right but no obligation to take ownership of what they have bought [9]. In this way, users have the opportunity to reserve their VMs by paying a reduced price in advance, and service providers benefit from an accurate estimation of their future needs. One of the cloud's leading service providers, Amazon Web Services (AWS) offers 12-month and 36-month pre-order VMs to its service users, which is called *Reserved Instances*. AWS reserved instances make it possible for users to attain their requested VMs at a relatively low price compared to on-demand instances, and the longer the contract, the lower the VM cost.

The key question is, is it possible to have the freedom of action and the benefits of on-demand pricing models for service users at a lower price, and at the same time, provide the information needed by service providers to predict and prepare for future needs, the way futures contracts and options contracts do?

Paper [10] introduces a broker role that acts as a mediator between service providers and service users. In this case, brokers buy 12-month or 36-month packages and divide them into smaller chunks and provide them to the service users. In [11], buyers who have consumed part of their purchased 12-month or 36-month packages can repackage their service surplus and sell it to demanded service users. Both these mediators, brokers and reseller buyers, create a secondary market and take the role of service providers for this new ecosystem.

In cloud primary markets, service providers directly communicate with their service users and offer their

services mainly through either on-demand mechanism or futures and options contracts. The advantages of the on-demand method for users are the flexibility of service usage time and cost management. One of the disadvantages of this method is the relatively high price of the offered services. Another disadvantage is that service providers cannot obtain the necessary information that they need to predict the future needs of the market. In contrast, in futures and options contracts, service providers can accurately predict the market's future demands, and service users benefit from much lower prices than on-demand ones. But the flexibility of service time and cost management are not available in futures and options contracts. Cloud secondary markets are created in response to the challenges of cloud primary markets to benefit from the advantage of on-demand methods, as well as futures and options contracts. The characteristics of these two environments, cloud primary markets and cloud secondary markets, are completely different from each other. In primary markets, resources are available to service providers, and if service providers are not able to sell their services, they can shut down inactive servers and save on their running costs. In cloud secondary markets, brokers and reseller buyers have attained their resources from service providers of the cloud primary market in the form of timed packages. As unsold packages do not transfer to the next interval, the mediators, brokers and reseller buyers need to sell their packages to service users as much as possible, or else, these packages will perish.

In this paper, we focus on such secondary markets in which mediators, brokers and reseller buyers, play the role of service providers to provide VMs to service users. To avoid confusion in the rest of this paper, we will use the term "service provider" instead of a mediator, broker, or reseller buyer, to indicate the secondary market's service providers. In such markets, we need to consider the time-criticality and perishability of VMs, as if these VMs are not traded in each round, they will not transfer to the next rounds. These VMs will perish, and their salvage value will become zero if they are not used or traded. Given the many-to-many relationships between service providers (sellers) and service users (buyers) in these secondary markets, double auction mechanisms are of the best choices to meet our needs. In such double auction mechanisms, sellers and buyers submit their asks and bids, respectively, to an auctioneer and the auctioneer matches them up. Time-criticality, perishability, and task priority have not received enough attention in the studies of double auction mechanisms for IaaS in cloud ecosystems. Not considering the perishability and time constraints of providers' services and users' demands can drastically impact resource allocation, as well as the overall social welfare. Double auction mechanisms fall within the category of dynamic pricing. Compared

to fixed-price-scheduling mechanisms, dynamic pricing approaches provide higher resource allocations and maximize social welfare (a.k.a agents' utility) for both service providers and service users of the cloud. In dynamic pricing models, double auctions are suitable mechanisms that benefit from high clarity and relatively low computational complexity and can appropriately deal with self-interested and strategic behaviours of cloud service agents (providers and users) [12, 13].

The current double auctions in cloud ecosystems have not sufficiently addressed the effect of perishability and time-criticality on resource allocation and system utility [14]. The time-criticality concept is important not only for service providers but also for service users. If service users do not receive requested resources before their deadline, they are subject to losing their profits. This issue then causes service providers to lose unsatisfied service users as well. In short, time-criticality is rooted in the perishable nature of the services and resources in the secondary market of IaaS ecosystems.

From another perspective, not all users and tasks have the same level of priority and urgency. For instance, administrative tasks should receive high priority to guarantee that the whole process remains functional. On the other hand, cloud service providers need to be able to offer different quality of service, e.g. platinum, gold or silver membership services, based on the demands and budgets of cloud service users. In all of these cases, priority, as an essential parameter in cloud computing environments, in general, and in IaaS secondary markets in particular, should be considered.

In [15, 16], and [17] the concept of time-criticality and perishability have been studied for perishable services, and in [18] the perishability concept has been studied for perishable goods. The VMs in IaaS secondary markets have unique characteristics and do not fit perfectly in either the perishable service or perishable goods categories. Hence, we cannot directly apply perishable service or perishable goods methods to the IaaS secondary market ecosystem, and we have to tailor the required algorithms and formulas.

To the best of our knowledge, in the double auction mechanisms of IaaS secondary markets that facilitate resource allocation and price-scheduling, the time-criticality, task priority and perishable nature of resources and services were not considered thoroughly. Despite all the value added features of proposed double auction mechanisms, considering the perishability and the time-criticality of cloud resources and services can improve the overall resource allocation efficiency and agents' utility. In classical double auction approaches, the main goal is to find the best pair of ask and bid matches, even though potential trades and of invested resources maybe lost. Based on the stochastic nature of cloud service demands

and the perishability of cloud resources and services, finding a balance between supply and demand to increase the trade success rate is complicated. This paper leverages the positive features of current double auctions while considering the perishability and time-criticality of resources and services to increase the number of successful trades. It increases the level of resource allocation, as well as the overall social welfare.

In short, the contribution of the present article is summarized in the following:

- The state-of-the-art price-based double auctions do not thoroughly address the time-criticality of the participants' tasks. In this paper, a priority-based dynamic double auction model is proposed which considers the perishability and time constraints of the involved parties' activities in IaaS secondary markets.
- Based on the suggested model, we propose a mechanism that improves the successful-trade rate by increasing the matching rate factor and the overall utility. To prioritize the time-critical tasks, we have defined ask and bid satisfiability factors to measure the criticality of the sellers' asks and the buyers' bids. Moreover, a defer rate is defined to delay the matching process to accept any high-priority or urgent incoming task without sacrificing the performance in resource allocation and traders' profits.
- By running an extensive number of simulations in different scenarios, it has been shown that the proposed model is superior to the price-based model in terms of resource allocation and social welfare.

The rest of this paper is organized in the following manner: in "Related work" section the state-of-the-art studies in the field of cloud resource allocation and price-scheduling have been reviewed. Next, the system model and problem statement are explained. In "Proposed approach: Priority-based dynamic online double auction mechanism (PB-DODAM)" section, the Priority-based Dynamic Online Double Auction Mechanism (PB-DODAM) is thoroughly explained, followed by its priority-based allocation algorithms and price-scheduling mechanisms. In "Experimental results" section, the experimental results are provided and the characteristics of the proposed mechanism in different scenarios are analyzed. Finally, the paper is concluded by discussing the proposed mechanism's features and research directions for future work.

Related work

The auction is one of the most prevalent economic mechanisms that is widely used for resource allocation and price-scheduling in multi-agent environments, which

are proposed in various types [19]. Single-sided, double-sided, combinatorial auctions and their subsets have been used in cloud ecosystems for resource allocation and price-scheduling [15]. The following is a review of some of the recent papers in price-scheduling and resource allocation that use diverse types of auctions in cloud environments. We have classified these studies into single-sided, double-sided, and combinatorial auction groups.

Single-sided auction models

Kong et al. presented an adaptive VM algorithm for resource scheduling which uses a single-sided auction mechanism and considers a number of factors, including network bandwidth and auction deadlines [20]. Cloud property evaluation, VM configuration and finally auction payment are the steps of the suggested approach in this paper. These steps are designed to sort the users' bids in the auction's time-frame and to determine winning users. Moreover, it calculates payments and confirms the values received by the auctioneer. As the algorithm was designed to maximize the cloud service providers' profits, it suffers from the potential for monopolistic behaviour. Furthermore, the target is to increase resource utilization without a clear policy for price-scheduling.

To manage a dynamic VM allocation, Nejad et al. proposed an integer programming model where service providers specify the market price and provide the requested VMs to the winning users [21]. As they have relied upon from a single-sided auction model in their proposed approach, their mechanism is at risk from monopolistic providers' behaviours.

None of the above work has addressed time-criticality and task priority in single-sided auction models. The following paper uses a single-sided auction model to address perishability and time-criticality. Their environment is different from IaaS secondary markets, which are the main focus of our paper.

Nadjaran Toosi et al. proposed a prompt adoption mechanism, using a dynamic price-scheduling approach to keep the balance for cloud resources in supply and demand markets [14]. This paper relied on a single-sided auction mechanism and is the main paper that uses the concept of perishable goods in providing price-scheduling mechanism for resources at the cloud data-centre level. This work calculates reserve prices based on electricity costs and data-centre Power Usage Effectiveness (PUE) for cloud data-centre resources. The proposed method is almost incentive-compatible while obtaining a nearly optimal benefit for cloud service providers, making the mechanism biased towards the providers. Moreover, as the research focuses on the data-centre resource level, it does not provide any solution for cloud offered services or engaged resources at the service level.

Double-sided auction models

Kumar et al. provided detailed research on different types of double auctions, considering a variety of challenges and obstacles to increasing the overall performance [22]. A truthful resource allocation and a price-scheduling mechanism are proposed as a truthful multi-unit double auction model (TMDA). The model tries to provide an incentive-compatible double auction mechanism while maintaining acceptable levels of other double auction pillars.

Wang et al. proposed a fitness-enabled auction mechanism as a new approach in cloud resource allocation that secures the performance traits for both service providers and service users [23]. Compared to continuous double auctions that do not benefit from the fitness idea, the fitness-enabled auction mechanism outperforms similar approaches in terms of resource allocation efficiency. In this paper, the high-level cloud services are more likely to receive higher quality resources.

Yashwant et al. proposed a double auction mechanism to balance energy-efficient resource allocation and service providers' social welfare, whereas most of the current research does not consider both sides [24]. Their proposed solution is a truthful double auction mechanism based on the Vickrey-Clarke-Groves (VCG) algorithm. Using a multi-dimensional bin-packing approach increased the performance of the algorithm compared to the inherent NP-hard computational complexity of similar mechanisms. Social welfare maximization is biased towards service providers and is not equally fair to both cloud service providers and cloud service users.

Wei et al. presented a resource allocation mechanism using the imperfect-information Stackelberg game model [25]. In this paper, the available historical demand records are used in a hidden Markov model to predict cloud service providers' current prices for their offered resources. Their dynamic price predictions are utilized in the proposed imperfect information Stackelberg game model (IISG). This approach also provides an optimal price-scheduling mechanism for service providers to maximize their social welfare. Since the IISG model yields an NP-hard problem, the authors consider the price-scheduling and resource allocation mechanisms from the service providers' perspective, which could impose monopolistic behaviour.

A P2P cloud or peer-assisted cloud is a decentralized type of cloud environment that utilizes a number of dissimilar computers [26]. Paper [27] put forward a hierarchical double auction algorithm for peer-assisted cloud ecosystems. In this paper, a non-cooperative game is proposed in which the agents compete over bandwidth allocation. The proposed algorithms cannot be used directly in client-server cloud architectures.

Formulating the VM provisioning problem and calculating the actual number of VMs are not trivial tasks. Paper [28] is one of the latest studies that utilize Lyapunov optimization techniques to design a cost-aware algorithm to calculate the accurate number of VMs which are requested by users in cloud environments. We can consider this paper as the enhancement of classical double auction mechanisms in IaaS environments that provides a VM allocation mechanism, as well as a price-scheduling model. It selects the second-price auction mechanism for calculating the VM price and the allocation phase. The paper asserted that the original VCG mechanism does not benefit from budget-balancing. Hence, their offered C-DSIC mechanism suffers from a lack of budget-balancing. Therefore, their proposed C-BIC mechanism uses a Bayesian incentive-compatible approach to alleviate this problem. As a result, the proposed model fulfills the individual rationality property of double auctions, while supporting the budget-balancing feature to an acceptable extent. The feasibility and efficiency of the proposed algorithm are represented and proven by the simulation.

The perishability and time-criticality of cloud resources and cloud services, as well as the task priority, were not addressed in any of the above-mentioned double auction models in cloud ecosystems.

Paper [29] is one of the pioneering work that has added the concept of time to a classical single-valued double auction mechanism. This paper uses Maximum-weighted Bipartite Matching Allocation (MBM Allocation) for resource allotment and thereby strives to prioritize tasks that are closer to the deadline to maximize the overall social welfare. The proposed mechanism considers a single unit of trade. Therefore, it does not contain the required generality for cloud-like environments that the trade of multiple units of resources happens regularly. Moreover, in this work, the agents who participate necessarily have dissimilar reports in the auction. This is not a true assumption for environments such as cloud ecosystems that some service providers and some service users with the same specifications can join the auction. The proposed mechanism cannot handle matching such cases, and this mainly goes back to the inherent limitations of augmentation techniques in graph theory, which authors utilized to implement their algorithm. Moreover, the order of implementation of MBM allocation and its offered Min-Max payment is $O(n^3)$, which is not appropriate for ever-growing capacity of environments such as cloud ecosystems. Furthermore, this paper does not regard the tasks with higher priorities that need to be considered in cloud ecosystems.

Miyashita et al. proposed an online double auction mechanism for perishable goods that considers time-criticality of the products in trades. The focus of this paper is on reducing the trade failure rate to increase the

efficiency [18]. Moreover, this research proves that considering the time-criticality of the products improves the agents' welfare as well. As paper [18] works on perishable goods, its methodologies cannot be directly applied to cloud resources and services in IaaS secondary markets.

Combinatorial auction models

In combinatorial auction models for clouds, the components that compose the cloud, e.g. CPU, storage, memory, and network, are considered to define different cloud types. In paper [30], a combinatorial one-to-many auction mechanism offered as Combinatorial Auction-Linear Programming (CA-LP) and Combinatorial Auction-Greedy (CA-GREEDY) provides a higher efficiency and utility, compared to fixed-price-scheduling. The single-sided auction mechanism in [30] has the risk of monopolistic behaviour which is one of the drawbacks of any one-sided auction. To resolve this problem, Samimi et al. in [13] offered a combinatorial double auction resource allocation (CDARA) that uses price averages for the final trade prices. CDARA suffers from the lack of truthfulness in its proposed mechanism. Paper [31] proposed A Fair Multi-attribute Combinatorial Double Auction Model (FMCDAM), which is founded on [13] and [32], for resource allocation in cloud environments. This paper focuses on the efficiency in allocation and fairness in price-scheduling, using a greedy allocation method and average price-scheduling respectively. In fact, using the service providers' reputation and the Quality of Service (QoS) as the involved parameters in the allocation and the pricing of the proposed method distinguishes FMCDAM from similar attempts and studies. Chen et al. offered greedy-based combinatorial double auction algorithms for homogeneous and heterogeneous platforms to increase the allocation efficiency and social welfare [33]. However, this paper does not include any payment mechanism in the proposed double auction.

G. Vinu et al. offered a combinatorial auction mechanism in [34], which provides the possibility of choosing different resources from various cloud providers or cloud vendors. Providing arbitrary packages of cloud resources from a diverse range of cloud providers and cloud vendors is the main distinguishing factor of the above-mentioned research. T. Bahreini et al. provided a two-level resource allocation and price-scheduling mechanism in edge computing systems [35]. The offered mechanism is based on cloud or edge resource allocation, and their price evaluation is based on a multi-unit combinatorial auction, as well as a position auction approach. In position auction, users have dissimilar preferences for using each resource, and the preference definition is based on the average distance between the resources and the service users. However, the preference definition does not include the task priority of the offered or requested services.

Focusing on both sellers' and buyers' benefits, paper [36] applies a double-sided combinatorial auction (DCA) for resource allocation in Transparent Computing (TC), which is considered the future of network computing. The idea is to maximize the social welfare while keeping the other double auction pillars at acceptable levels. Although the network bandwidth as the main parameter of TC is inherently perishable, the research did not consider its perishability. This paper only focuses on increasing the bandwidth fairness by prioritizing the tasks based on available historical data. To simplify the process, a stochastic modelling was preferred over probabilistic ones. The idea is to increase the chances of winning for the low-bid participants in the auction, while not considering time-criticality and perishability to decrease the trade failure rate.

Paper [37] is one of the more recent studies that offer a game-based combinatorial double auction mechanism to model a relationship between Infrastructure Providers (INP) and Service Providers (SP). As cloud ecosystems deal with several different components, it is a multi-dimensional combined resource environment. Traditional cloud price-scheduling models are mostly based on the average price of resources engaged in the process. This paper proposed a more accurate pricing approach by offering a combinatorial double auction mechanism based on an incomplete information game theory. Both INPs and SPs are self-interested entities and their goal is to maximize their profits, and they are not aware of each other's true valuation. Compared to the traditional cloud pricing model, this study utilized an incomplete information game to provide a more accurate price-scheduling mechanism. Using a Harsanyi transformation, this research can convert the proposed combinatorial double auction model into a complete, but imperfect information game mechanism.

None of the above combinatorial auctions consider the time-criticality and the task priority in the IaaS secondary markets, which is the focus of the current paper.

In this paper, we will show that using time-criticality, perishability, and task-priority can drastically improve resource allocation and the overall social welfare in cloud secondary markets. Our proposed model, Priority-based Dynamic Online Double Auction Mechanism (PB-DODAM), is described in the following sections.

System model and problem statement

The ordinary double auctions for clouds are price-based, and their main objective is to find the best pairs of asks and bids to be matched. In contrast, in conditions where passing the time decreases the chances of successful trades, the time-criticality and task priority should be taken into consideration. These two factors are not well-addressed in the current double auction models

for cloud ecosystems. Our proposed approach, priority-based dynamic online double auction mechanism (PB-DODAM), offers a double auction model for the IaaS secondary markets in cloud environments that takes time-criticality and task-priority factors into account.

In our priority-based dynamic online double auction model, there are a number of auction rounds which are called time-slots. The time-slot is determined as a discrete parameter, which is indicated by t , and each time-slot contains a number of minutes as our designated time unit and is represented by t' . We also have the concept of period, represented by p , which means the length of time that an agent has been on the market. In other words, the period is the time interval between the entrance and exit of an agent in the time-slot t and during the available t' minutes in that time-slot. Cloud service providers as sellers (S) and cloud service users as buyers (B) play the role of agents in the proposed model. The proposed model designed for a generic cloud environment that offers Infrastructure as a Service (IaaS) to its users in secondary markets and the agents are willing to trade some VMs as their trading units in the market. To simplify the experiment conditions, we consider that agents trade one type of VMs. Every service provider can offer a number of VMs, defined as α VMs, where α is an integer number, e.g. 12 VMs. Similarly, every service user can order a number of VMs. In every auction round, a number of VMs will be traded among the agents and at the end of each round the market will be cleared. Table 1 represents the notations which are used in this paper.

Every agent at time-slot t within the period p has its specifications which are called *type*, as defined in Definition 1.

Definition 1 (Agent type definition) : *Each agent i is defined with its type θ_i as follows:*

$$\theta_i = (v_i, q_i, a_i, d_i) \quad (1)$$

In Eq. 1, v_i represents a single unit valuation, whereas q_i identifies the number of VMs which agent i wants to trade. The agent's arrival and departure times are indicated by a_i and d_i , considering that $d_i > a_i$. Moreover, both a_i and d_i occur within a time-slot t . All of the above-mentioned parameters for the θ_i are non-negative numbers. The trading period $[a_i, d_i]$ for the agents is represented by p . Every agent can participate in the auction in a number of trading periods in different time-slots. In each trading period, the arrival time could be the time that an agent wants to trade or the time that the agent becomes aware of the auction. The departure time for service providers is the time that they receive their price, whilst for service users it is when they complete their payments. Considering zero to be the value that service users are willing to pay and ∞ to be

Table 1 Notations

Symbols	Descriptions
S	Cloud service providers (sellers)
B	Cloud service users (buyers)
t	Time-slot
t'	Time (within each time-slot)
α	Number of VMs ($\alpha \in \mathbb{N}$)
θ_i	Type of agent i
$\hat{\theta}$	Collection of all agents' types
$\hat{\theta}^t$	Collection of all agents' types in time-slot t
$\hat{\theta}^{\leq t}$	Collection of all agents' types from the beginning till the end of time-slot t
v_i	The single unit valuation by agent i
q_i	The total VMs' quantity that agent i wants to trade
a_i	The agent's arrival time
d_i	The agent's departure time
p	Agent's presence period in a time-slot
$U_i^{\hat{\theta}^t}$	Seller's i utility at time-slot t
$U_j^{\hat{\theta}^t}$	Buyer's j utility at time-slot t
P	Payment rule
s_{ij}	Seller's obtained value from the auctioneer
b_{ij}	Buyer's payment value to the auctioneer
Q	Allocation rule
M^t	Matchable pairs of asks and bids at time-slot t
$\sigma_i(\hat{\theta}^t)$	Seller's ask satisfiability in time-slot t
$c_i^p(\hat{\theta}^t)$	Seller's ask criticality in time-slot t and time t'
$\sigma_j(\hat{\theta}^t)$	Buyer's bid satisfiability in time-slot t
$c_j^p(\hat{\theta}^t)$	Buyer's bid criticality in time-slot t and time t'
τ	Satisfiability threshold
$\phi(t)$	Market price
$e_i^p(t)$	Agent i 's defer rate
ε	Defer rate threshold

the value that service providers are willing to receive provides an empty window when no trade happens. As in the real world, the agents are self-interested and do not reveal their information, so we have considered the agents' types to be private data. Agents can change their types in each time-slot, whereas they cannot enter the same bid if they depart once in that time-slot, but they can participate in the next round.

In our proposed model, the service provider i offers their VMs at the arrival time a_i , and the salvage value of the unused VMs becomes zero unless the VMs are traded successfully before the departure time d_i . For the service provider i , v_i is the combination of production and opportunity costs for each VM, which disappears at the departure time if no trade happens. This imposes the risk

of not compensating the costs in the case of trade failure. On the other hand, the service user j evaluates the value of their received VMs as v_j if their designated task is accomplished within their time limit. a_j denotes the time when the service user j values VMs, and d_j represents the maximum time to receive their VMs, and there is no use in receiving them any moment later.

To simplify the proposed model, we have assumed that each agent can offer only one ask/bid in every time-slot t . All the types of agents in time-slot t are represented by $\hat{\theta}^t$, while $\hat{\theta} = \{\hat{\theta}^1, \hat{\theta}^2, \dots, \hat{\theta}^t, \dots\}$ serves for all agents' types for the whole duration of the auction. $\hat{\theta}^{\leq t}$ represents the agents types from the first time-slot till the end of time-slot t . $\theta_i^p = (v_i^p, q_i^p, a_i^p, d_i^p)$ is an agent i 's type in the period p , whereas $\hat{\theta}_i^{t'} = (\hat{v}_i^{t'}, q_i^{t'}, a_i^p, d_i^p)$ denotes the type of agent i at time t' , while $t' \in [a_i^p, d_i^p]$.

The following describes the matching condition for every pair of the seller's ask and the buyer's bid:

Definition 2 (Matching requirements): *For every seller's ask in period p , when the seller's type is $\hat{\theta}_i^t = (\hat{v}_i^t, q_i^t, a_i^p, d_i^p)$ in time-slot t and for every buyer's bid in period p , when the buyer's type is $\hat{\theta}_j^t = (\hat{v}_j^t, q_j^t, a_j^p, d_j^p)$ in time-slot t , matching happens when the following conditions are fulfilled:*

- i. $\hat{v}_i^t \leq \hat{v}_j^t$
- ii. $[a_i^p, d_i^p] \cap [a_j^p, d_j^p] \neq \emptyset$
- iii. $q_i^p > 0$
- iv. $q_j^p > 0$

The first condition describes that the seller's ask should be less than the buyer's bid. The second condition suggests that sellers' and buyers' time-slots overlap. The third condition implies that the seller i should continue providing the requested services (i.e. the needed VMs) as long as it fulfills the buyer j 's request. The fourth condition implies that the requested resources from buyer j should be a positive value.

An online double auction mechanism is formulated as $M = \{P, Q\}$, where P denotes the payment rule and Q represents the allocation rule. The payment rule P is defined as $P^t = (s^t, b^t)$, where both s^t and b^t are non-negative numbers. $s_{i,j}^t$ depicts what seller i obtains from the auctioneer in time-slot t as a result of the trade with buyer j . Similarly, $b_{i,j}^t$ is what buyer j pays to the auctioneer in time-slot t as a result of the trade with seller i . In this paper, we have assumed that the proposed double auction is a Strong Balanced Budget, which means that the auctioneer neither gains nor loses any profit in the auction, and as a result, $s_{i,j}^t = b_{i,j}^t$.

Common approaches taken to calculate the utility function in double auctions does not consider the time-criticality and perishability [38]. Considering v_i to be the valuation of cloud agent i for a VM, the common utility function for seller i is calculated as $\sum_j (s_{i,j} - v_i Q_{i,j})$ and for buyer j , it is calculated as $\sum_i (v_i Q_{i,j} - b_{i,j})$. Both of these formulas are simple quasi-linear equations. For sellers, the utility function is the difference between what they receive from the auctioneer and what they evaluate as the price for their offered services. However, for buyers, the utility function is the difference between what they assessed as the value of their received services and what they actually pay the auctioneer. In our proposed priority-based dynamic online double auction approach, the utility function for seller i is calculated based on the deduction of untraded units' value from the common seller utility function. In other words, the priority-based sellers' utility is calculated based on the difference between what they receive and what they evaluate as the price of the traded units, minus the value of unsold VMs that are considered perished. The concept is illustrated by the next definition.

Definition 3 (Priority-based seller's utility) : *The following equation calculates the seller's utility in time-slot t which is obtained between the time of arrival and the time of departure:*

$$U_i^{\hat{t}} = \sum_{t' \in [a_i^p, d_i^p]} \left(\sum_{i \in S, j \in B} (s_{i,j} - v_i Q_{i,j}) \right) - \sum_{t' = d_i^p} \left(\sum_{i \in S, j \in B} (q_i - Q_{i,j}) v_i \right) \quad (2)$$

The first part of the Eq. 2 is similar to the classical quasi-linear utility function that we use for common online double auctions, whereas the second part of the equation calculates what sellers lose for their unsold VMs when they depart. q_i represents the total available VMs, whereas $Q_{i,j}$ denotes the number of VMs that the sellers have sold before the departure. The subtraction of these two values expresses the number of unsold VMs. In other words, the second part of the equation is the utility loss which was caused by the perishable nature of cloud services in IaaS secondary markets and is taken for granted in classical price-based approaches. This potential loss is the main reason to encourage sellers to modify and decrease their evaluation before their departure time.

The utility or the surplus that any buyer achieves is captured in the next definition.

Definition 4 (Priority-based buyer's utility) : *The following equation calculates the buyer's utility in time-slot t*

which is obtained between the time of arrival and the time of departure:

$$U_j^{\hat{t}} = \sum_{t' \in [a_j^p, d_j^p]} \left(\sum_{i \in S, j \in B} (v_i Q_{i,j} - b_{i,j}) \right) \quad (3)$$

In this paper, we assume that the engaged agents aim to maximize their utility and are risk-neutral entities. In Eq. 2, there is a trade-off between selling at high prices and having more successful trades. If a seller sells their VMs at higher prices, they can achieve more benefit. However, this can decrease the trade success rate and waste the unsold resources and services. On the other hand, lowering the prices increases the chance of successful trade while decreasing the direct benefit of the sold VMs. The trade-off between raising the prices and increasing the number of successful trades complicates the process of finding an appropriate trading price in each time-slot. On the one hand, time-criticality for sellers is defined as selling their VMs before the departure time and leaving no unsold resources or services as much as possible. On the other hand, the time-criticality for buyers is to accomplish their tasks before certain deadlines. In Eq. 3, buyers' profits increase by obtaining high-value VMs at lower costs. Considering buyers' tasks time-criticality and attempting to gain more profits by bidding for lower prices makes it difficult for buyers to find appropriate price-scheduling strategies.

Based on the defined sellers' and buyers' concerns, the utility maximization objective in our model is explained in the following definition.

Definition 5 (Priority-based utility maximization) : *In an online double auction $M = \{P, Q\}$, where P is the payment rule and Q is the allocation rule, the utility maximization function is responsible for selecting the payment and allocation rules that maximize the overall benefits. The following formula calculates the utility maximization in time-slot t which is obtained among the intersection of agents' presence in the time-slot:*

$$U_i^{\hat{t}} = \sum_{t' \in ([a_i^p, d_i^p] \cap [a_j^p, d_j^p])} \left(\sum_{i \in S, j \in B} (v_j - v_i) Q_{i,j} \right) - \sum_{t' = d_i^p} \left(\sum_{i \in S, j \in B} (q_i - Q_{i,j}) v_i \right) \quad (4)$$

In the first part of Eq. 4, the social welfare maximization is defined through $(v_j - v_i) Q_{i,j}$, and the second part, $(q_i - Q_{i,j}) v_i$, represents the benefit loss which has come from the untraded VMs.

No online double auction mechanism can optimize individual rationality, budget-balancing, truthfulness, and computational efficiency at the same time [39], and our approach is no exception. The ideal case is designing a strategy-proof mechanism that satisfies budget-balancing and computational efficiency. Green et al. [40] have proved the seminal impossibility of this setting and have asserted that no double auction can optimize both budget balance and efficiency at the same time. Myerson et al. in [41] have proved a more general version of this theorem. Our proposed mechanism satisfies the individual rationality, budget-balancing, and truthfulness, while trying to increase the computational efficiency. Sellers' utility, buyers' utility, and utility maximization guarantee the individual rationality property of our proposed online double auctions. Satisfying the strong balanced budget feature means the auctioneer receives or loses no benefit in any trade. Truthfulness implies that the agents have no incentive to misreport their types, which means there is no reason to falsely report the arrival or departure time of the agents. Increasing the successful trade ratio, as well as reaching for a better allocation efficiency, are the main goals of the current research, which can be achieved by using the utility maximization function. In short, the utility maximization function verifies the computational efficiency increment in our approach. At the same time, our proposed approach satisfies the individual rationality, budget-balancing, and truthfulness features of online double auction mechanisms.

In our proposed mechanism, we have assumed that the auctioneer does not play the role of short position which is known as feasibility feature. In the next section, we will discuss our proposed approach in more detail.

Proposed approach: Priority-based dynamic online double auction mechanism (PB-DODAM)

Due to the self-interested human tendency to try to maximize profit, sellers tend to raise their asks while buyers tend to lower their bids. This rational strategy can boost the agents' benefit as long as they do not face the risk of losing the trades. In durable goods markets, trade failure in each round can be proceeded with the success in the next round, while this is not the case for perishable goods or services. If service providers do not trade their VMs, their resources will be perished for that time-slot. For this reason, we need to consider perishability, time-criticality, and task-priority factors in our mechanism. In general, any online double auction consists of two major phases which are allocation and pricing mechanisms, and we will discuss these two mechanisms in our proposed approach in the proceeding sections.

Proposed allocation mechanisms

Maximizing the social welfare is the main objective of allocation mechanisms in static durable goods markets which is achieved by matching higher users' bids with lower sellers' asks. The summation of differences between all matched pairs of asks and bids calculates the social welfare in such markets. The maximal social surplus is the result of an ideal scenario which matches all existing asks and bids in the market. Allocation efficiency is one of the most important metrics in a competitive equilibrium to evaluate the effectiveness of any allocation mechanism. In our proposed method, matching rate is considered to be an indicator for the allocation efficiency. The matching rate is the result of splitting the current social surplus by the maximal social surplus.

To maximize the allocation efficiency in double auctions for durable goods in static markets, the sellers' asks should be sorted in ascending order, whereas the buyers' bids should be sorted in descending one. The next phase is to match the lower asks with higher bids until there is no unmatched pairs left in the market. This allocation approach is called the price-based allocation mechanism.

The price-based allocation mechanism works the best in conventional spot markets for durable goods or services. In these markets, failure in trading units in one round causes no harm to the salvage value of unsold units of goods or services, and they can be traded in future rounds. On the contrary, in the market for the perishable goods or services, such as IaaS secondary markets in cloud environments, the improvement in the successful trade rate should be considered along with the social surplus growth.

To increase the number of successful trades, one of the current approaches is to match high value bids with high value asks that are not above the bids valuation, and low value asks with low value bids that are not below the asks valuation. Although this approach can increase the allocation efficiency, it cannot tackle the perishability problem and prevent the loss that happens due to the trade failures. This is why we have considered *criticality* as a new parameter that can evaluate the trade urgency of offered productions or services in online double auctions. The role of criticality is to increase the trade chances of unsold units that arrive closer to their departure time and to increase the successful trade rate.

In this paper, M^t denotes all matchable pairs of asks and bids at time-slot t . At this time-slot, the total quantity of matchable bids with the seller i 's ask $\hat{\theta}_i^t$ is equal to $\sum_{(\hat{\theta}_i^t, \hat{\theta}_j^t) \in M^t} q_j^t$. At the same time-slot, for each buyer j , the total matchable asks' quantity is equal to $\sum_{(\hat{\theta}_i^t, \hat{\theta}_j^t) \in M^t} q_i^t$. The arrival time and the departure time for the seller i at time-slot t within period p are a_i^p and d_i^p , respectively. The seller i 's slack time in time t' is represented by $d_i^p - t'$

in which the offered VMs will be wasted if they are not traded after the departure time. To define the overall status of each ask in each time-slot, we define the seller's ask satisfiability in time-slot t in Definition 6.

Definition 6 (Seller's ask satisfiability) : *The satisfiability of seller's i 's ask at time-slot t is calculated as follows:*

$$\sigma_i(\hat{\theta}^t) = \sum_{(\hat{\theta}_i^t, \hat{\theta}_j^t) \in M^t} \left(\frac{q_j^t}{\sum_{(\hat{\theta}_i^t, \hat{\theta}_j^t) \in M^t} q_j^t} \right) \quad (5)$$

In Eq. 5, the numerator denotes the total buyers' bids that exist in the market and are matchable with the ask $\hat{\theta}_i^t$ of seller i in time-slot t . The denominator represents the total available quantity of sellers' asks in the market that are matchable with buyer j 's bid at the same time-slot. If the result of the fraction is bigger than 1, the demand quantity is greater than the supply. This condition increases the competition among the buyers for receiving their needed services. On the other hand, if the result of the fraction is less than 1, the supply quantity is greater than the demand, which increases the competition among the sellers to sell their services. For instance, if the total quantity of the market demand is 100 while the total number of available VMs is 20, the result of the fraction is $100/20 = 5.0$, which means that for each requested VM, there is 0.2 available VM. In this case, the competition is among the buyers to obtain their needs. Similarly, if the total quantity of the market demand is 10 VMs while the total number of available VMs is 20, the result of the fraction is $10/20 = 0.5$, which means that for each VM request, there are 2 available VMs. In this case, the competition is among the sellers to sell their available VMs.

In short, σ_i represents the overall asks' conditions in each time-slot and demonstrates how safe or critical any ask's condition is. The bigger σ_i provides higher satisfaction for the sellers, whereas the smaller σ_i brings less satisfaction to the sellers. Apart from this important factor, we should take the time limit into account, as when we get closer to the departure time, the chances of finding an appropriate match decrease and the possibility of trade failure increases. Definition 7 combines sellers' satisfaction and time-criticality features.

Definition 7 (Seller's ask criticality) : *The criticality factor for seller i 's ask at time-slot t within period p is calculated as follows:*

$$c_i^p(\hat{\theta}^t) = \frac{1}{(\sigma_i(\hat{\theta}^t))(d_i^p - t')} \quad (6)$$

In Eq. 6, seller i 's ask satisfiability and slack time are located in the denominator. When the satisfiability

becomes smaller and the t' becomes closer to the departure time, the denominator decreases and the whole fraction, which is the ask criticality, increases. In this case, a mechanism should be defined to enhance the trade possibility by increasing the ask's priority for the trade.

Similarly, in the next two definitions, we will discuss buyer's bid satisfiability and buyer's bid criticality.

At time-slot t , the total quantity of matchable asks with the buyer j 's bid $\hat{\theta}_j^t$ is equal to $\sum_{(\hat{\theta}_i^t, \hat{\theta}_j^t) \in M^t} q_i^t$. At the same time-slot, for each seller i , the total matchable bids' quantity is equal to $\sum_{(\hat{\theta}_i^t, \hat{\theta}_j^t) \in M^t} q_j^t$. The arrival time and the departure time for the buyer j at time-slot t within period p are d_j^p and d_j^p , respectively. Buyer j 's slack time in time t' is represented by $d_j^p - t'$, and it is mandatory to receive the requested VMs before the departure time to accomplish the required tasks. To define the overall status of each bid in each time-slot, we define the buyer's bid satisfiability in time-slot t in Definition 8.

Definition 8 (Buyer's bid satisfiability) : *The satisfiability of buyer j 's bid at time-slot t is calculated as follows:*

$$\sigma_j(\hat{\theta}^t) = \sum_{(\hat{\theta}_i^t, \hat{\theta}_j^t) \in M^t} \left(\frac{q_i^t}{\sum_{(\hat{\theta}_i^t, \hat{\theta}_j^t) \in M^t} q_j^t} \right) \quad (7)$$

In Eq. 7, the numerator denotes the total sellers' asks that exist in the market and is matchable with the bid $\hat{\theta}_j^t$ of buyer j in time-slot t . The denominator represents the total available quantity of buyers' bids in the market that are matchable with seller i 's ask at the same time-slot. If the result of the fraction is bigger than 1, it represents that the supply quantity is more than the demand. This situation increases the competition among the sellers to sell their services. On the other hand, if the result of the fraction is less than 1, the demand quantity is greater than the available services. This condition increases the competition among the buyers to receive their needed services. For instance, if the total number of available VMs is 50 while the total quantity of the market demand is 10, the result of the fraction is $50/10 = 5.0$, which means that for each requested VM, there are 5 available VMs. In this case, the competition is among the sellers for selling their available VMs. Similarly, if the total number of available VMs is 50 while the total quantity of the market demand is 100 VMs, the result of the fraction is $50/100 = 0.5$, which means that for each VM request, there is 0.5 available VM. In this case, the competition is among the buyers to obtain their needs.

In short, σ_j represents the overall bids' conditions in each time-slot and demonstrates how safe or critical any bid's condition is. The bigger σ_j has a higher satisfaction for buyers, whereas the smaller σ_j brings less satisfaction

to buyers. Apart from this important factor, we should consider the time limit, as when we get closer to the departure time, the chances of finding an appropriate match decrease and the possibility of trade failure increases. Definition 9 combines buyers' satisfaction and the time-criticality features.

Definition 9 (Buyer's bid criticality) : *The criticality factor for buyer j 's bid at time-slot t within period p is calculated as follows:*

$$c_j^p(\hat{\theta}^t) = \frac{1}{(\sigma_j(\hat{\theta}^t))(d_j^p - t')} \quad (8)$$

In Eq. 8, buyer j 's bid satisfiability and slack time are located in the denominator. When the satisfiability becomes smaller, and the t' becomes closer to the departure time, the denominator decreases, and the whole fraction, which is the bid criticality, increases. In this case, a mechanism should be defined to enhance the trade possibility by increasing the bid's priority for the trade.

In price-based allocation mechanisms, the priority in matching bids and asks is to pair the higher bids with lower asks. To implement this strategy, we can sort the negative asks' valuation, $-\hat{v}_i^t$, and the negative reciprocal of bids' valuation, $-1.0/\hat{v}_j^t$, both in descending order, and pair them in a sequence. In our proposed allocation mechanism, the satisfiability and the criticality define the ask's and bid's priority for matching, which significantly change the whole allocation and pricing mechanism.

Figure 1 depicts the overall PB-DODAM allocation scheme and the relationship among service providers (sellers), service users (buyers), and an auctioneer in an IaaS secondary market. As shown in the figure, there is a number of asks that start from $\theta_1 = (v_1, q_1, a_1, d_1)$ to $\theta_i = (v_i, q_i, a_i, d_i)$ and also a number of bids which start from $\theta'_1 = (v'_1, q'_1, a'_1, d'_1)$ to $\theta'_j = (v'_j, q'_j, a'_j, d'_j)$. Each of these asks and bids arrives to and departs from the auction in different times at time-slot t within period p . Some of these asks or bids may arrive earlier and depart sooner than the others, while some asks have joined later, or some bids have more time to fulfill their needs. This variation defines a different criticality for each attended agent and shows which ones need to be paired as quickly as possible. Moreover, it represents which agents can defer and wait for potentially better matches. The battery sign near each agent depicts the time-criticality, and if it has more battery bars in green, it means there is more time to wait for better deals. Less battery bars imply that the agent has an urgent situation to find its match, and if it cannot find the match, it will be perished. Our proposed mechanism wants to consider the perishable and priority-based nature of cloud resources or services in IaaS secondary markets

to increase the number of successful trades. This will lead to an increase in the overall utility and profitability of the system.

This conceptual view is formulated in Algorithms 1 and 2, by applying the defined satisfiability and criticality concepts.

Algorithm 1 Ask's and Bid's Priority Determination

Input Stage: Set of active asks and bids, satisfiability threshold (τ)

Output Stage: Prioritized asks and bids based on agents' satisfiability

- 1: Adding arriving asks and bids at time-slot t in period p to the group of sellers (S), and buyers, (B), respectively and removing asks and bids that depart at the same time
 - 2: **for** each seller i **do**
 - 3: Calculating σ_i as the ask i 's satisfiability
 - 4: **if** $\sigma_i \leq \tau$ **then**
 - 5: $Ask(i)'sPriority \leftarrow c_i^p(\hat{\theta}^t)$
 - 6: **else**
 - 7: $Ask(i)'sPriority \leftarrow -\hat{v}_i^t$
 - 8: **end for**
 - 9: **for** each buyer j **do**
 - 10: Calculating σ_j as the bid j 's satisfiability
 - 11: **if** $\sigma_j \leq \tau$ **then**
 - 12: $Bid(j)'sPriority \leftarrow c_j^p(\hat{\theta}^t)$
 - 13: **else**
 - 14: $Bid(j)'sPriority \leftarrow -1.0/\hat{v}_j^t$
 - 15: **end for**
-

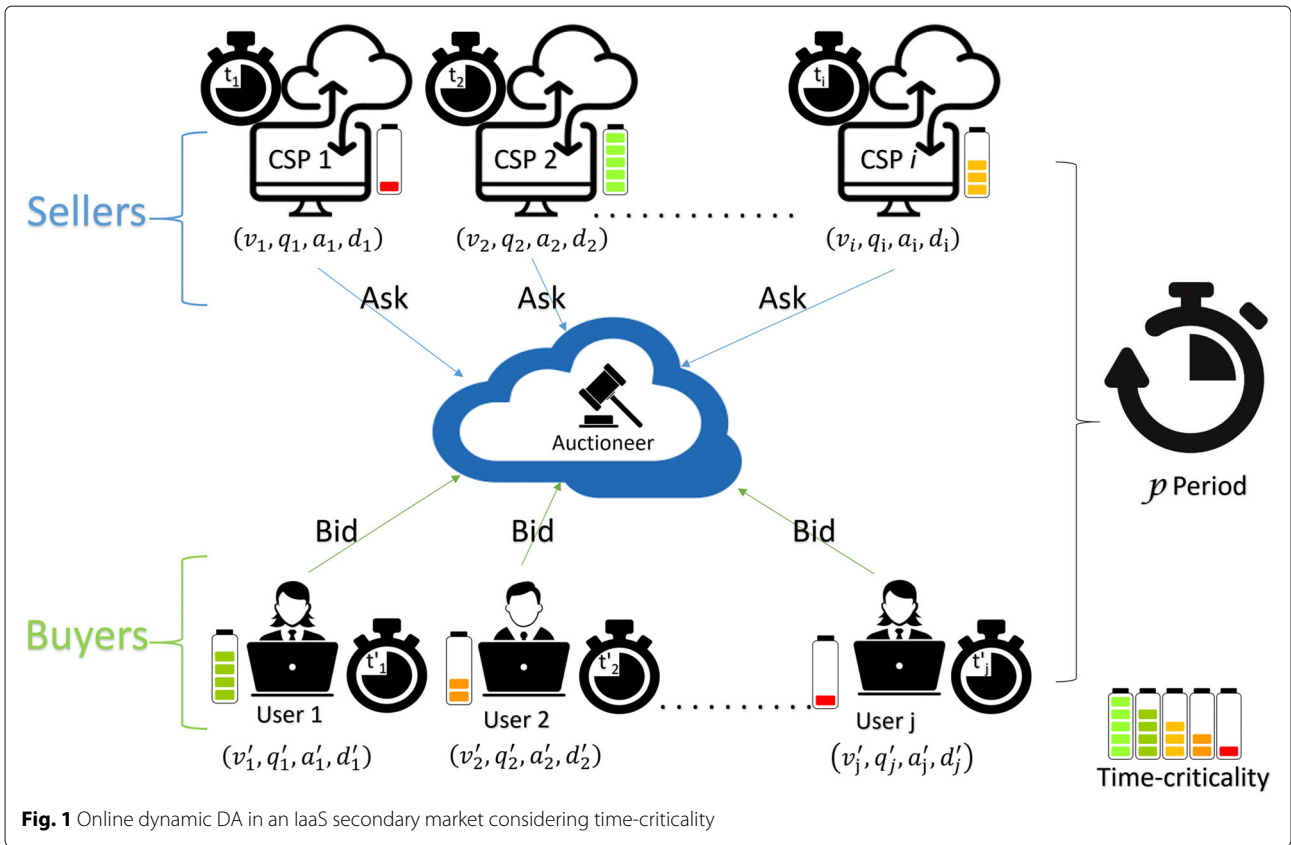
Ask's and bid's priority algorithm description

The following part defines the steps of Algorithm 1, *Ask's and Bid's Priority Algorithm* in more details:

Step 1: All agents' asks and bids are received and classified into the seller or buyer group in every time-slot while removing the departed asks and bids in the same time-slot.

Step 2: The satisfiability for every seller in every time-slot is calculated. If the satisfiability of seller i is less than a defined threshold, a priority will be assigned to the seller based on the calculated criticality value. Otherwise, the seller's status is not critical, so noncritical asks are separated, and the negative of the original valuation is assigned to them as their priority.

Step 3: Similarly, the satisfiability for every buyer is calculated in every time-slot. If the satisfiability of buyer j is less than a defined threshold, priority will be assigned to the buyer based on the calculated criticality value. Otherwise, the buyer's status is not critical, so the noncritical bids are separated, and the negative reciprocal of the original valuation is assigned to them as their priority.



In steps 2 and 3, if the agents' asks and bids are in critical status, their priority will be calculated accordingly. Otherwise, for noncritical asks and bids, we follow the same price-based allocation mechanism. We define the satisfiability threshold, τ , to distinguish between the critical and noncritical cases. An appropriate value for the satisfiability threshold is obtained through a numerous number of experiments using a multi-agent simulation environment that will be discussed thoroughly in "Experimental results" section.

In the next phase, which is the priority-based allocation algorithm, we start by sorting the asks and bids based on their calculated priorities to match them.

Priority-based allocation algorithm description

The following part defines the steps of Algorithm 2, *Priority-based Allocation Algorithm*, in more details:

Step 1: First, sort the prioritized asks in descending order based on their calculated priority precedence which is explained in Algorithm 1.

Step 2: Pick the first ask in the list that has the highest priority.

Step 3: Find all the machable bids in the bids list and update their priority.

Step 4: Sort these machable bids based on their updated priorities in descending order.

Algorithm 2 Priority-based Allocation Mechanism

Input Stage: Prioritized asks and bids based on agents' satisfiability

Output Stage: VMs allocation based on prioritized asks and bids

- 1: Sorting asks in descending order based on their updated priorities
- 2: **for** each seller i 's ask in time-slot t **do**
- 3: Find all machable buyers' bids with seller i 's ask and update their priorities
- 4: Sort the machable bids based on their updated priorities in descending order
- 5: **if** sorted machable bids \neq null **then**
- 6: pair ask i and highest priority bid j
- 7: remove ask i and bid j from the sorted asks and bids lists
- 8: **else**
- 9: move ask i to the unsatisfied list and move on to the next ask
- 10: **end for**

Step 5: If the sorted machable bids collection is not null, ask i is paired with the highest priority bid j , and both matched ask and bid are removed from the list. Then the next ask from the list should be selected to repeat

the whole process. If there is no matchable bid for the current ask, the ask should be moved to the unsatisfied list, and the next ask should be selected to repeat the aforementioned process.

Step 6: Step 1 to step 5 are repeated until no ask remains without matchable bids.

From a broader perspective, the price-based allocation mechanism is a subsidiary of our proposed approach for the case when the satisfiability threshold is set to 0. In other words, based on the Algorithm 1, if the satisfiability threshold is equal to 0, any bid or ask would be greater than zero and time-criticality is not considered. On the other hand, if we set the satisfiability threshold equal to ∞ , all asks' and bids' conditions would be considered as critical and the allocation mechanism would become a pure criticality-based mechanism.

Although the criticality-based allocation algorithm focuses on increasing the successful trade rate and improving the allocation efficiency, achieving less trade failures could potentially improve the overall surplus. In other words, the main goal of the proposed algorithm is to find a proper resource allocation mechanism that increases the efficiency and decreases the wasted resources or services. As a result, increasing the successful trade rate improves the overall social welfare. In comparison to the price-based approach, which only focuses on increasing economic productivity, our proposed criticality-based mechanism is more sustainable in terms of higher resource allocation efficiency. At the same time, it satisfies the rest of the features of online double auction mechanisms. In the following theorems, we prove that our priority-based allocation mechanism satisfies the individual rationality, budget-balancing, and truthfulness features of online double auction mechanisms.

Theorem 1 *The proposed PB-DODAM mechanism is truthful.*

Proof We provide a sketch of the proof for the truthfulness of the PB-DODAM mechanism. Self-interested entities allow themselves to report their types with dishonesty to gain as much interest as possible. Among the four parameters of each agent's type, misreporting the arrival and departure time has no use. As the arrival time a_i for the agents is the earliest time that they are willing to trade, reporting an earlier time has no logic. Reporting a later arrival time ($a'_i > a_i$), as well as the earlier departure time ($d'_i < d_i$), decreases the chances of matching pairs and increases the possibility of trade failures. Moreover, reporting later departure times brings no benefit to sellers and buyers, and jeopardizes the required finishing time, so buyers may receive their VMs when they cannot finish their designated tasks. Requesting more quantity ($q'_i > q_i$) from buyers is irrational as they should pay more

for unwanted services and demanding fewer resources endangers their task completion. From the sellers' side, offering more than available resources is not possible as the sold VMs need to immediately be delivered to service users. Offering fewer resources with the intention of creating false deficiencies to raise the price requires a high degree of knowledge of current markets. Moreover, it increases the computational complexity to find the exact amount that should be offered to maximize the profit. Even if we could find this optimal point, not selling to the full capacity is equal to wasting the resources and consumed electricity. In terms of valuation, there is no rational if a service user reports their valuation more than what they willingly want to pay ($v'_i > v_i$), as this decreases their profit. On the other hand, reporting the service user's valuation less than their actual dedicated budget decreases their matching chances and causes them to lose the trade. Similarly, if a seller asks for more than what they genuinely expect to receive, they may lose the trade. Based on the perishable nature of cloud resources and services, sellers can go below their initial valuation when they come closer to their departure time. This strategy can increase their chances for successful trades. The reason for this is if they do not sell their VMs, they will perish for that time-slot. For this reason, we do not consider reporting lower valuation from the service providers' side as misreporting, but instead a strategy to increase their successful trades. Therefore, it is reasonable that no agent misreports their type for strategic behaviours. The only modification that can happen is on service providers' valuation to not lose the trade and the salvage price of their VMs, which is not considered as misreporting. \square

Theorem 2 *The proposed PB-DODAM mechanism is incentive-compatible and individually rational.*

Proof PB-DODAM mechanism is a combination of P and Q , where the payment rule, P , is a real-valued function from the buyer to the seller on $[0, 1]^2$ with v_1 and v_2 values. The allocation rule, Q , is a mapping between $[0, 1]^2$ and $[0, 1]$. The value at (v_1, v_2) defines the probability of the trade. If the values of P and Q result from a Bayesian game with a pair of Bayesian-Nash parity strategies, then PB-DODAM mechanism (P, Q) can be considered *incentive-compatible*. PB-DODAM applies the following k -double auction allocation rule which results in equilibrium (P', Q') :

$$P(v_1, v_2) = \begin{cases} kP'(v_2) + (1 - k)Q'(v_1) & \text{if } P'(v_2) \geq Q'(v_1) \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

and

$$Q(v_1, v_2) = \begin{cases} 1 & \text{if } P'(v_2) \geq Q'(v_1) \\ 0 & \text{otherwise,} \end{cases} \quad (10)$$

The type v_1 seller's interim expected utility can be calculated for an incentive-compatible double auction mechanism (P, Q) as follows:

$$U_1(v_1; P, Q) = \int_0^1 [P(v_1, v_2) - v_1 Q(v_1, v_2)] dF_2(v_2) \quad (11)$$

The sellers' ex-ante expected utility is calculated as follows:

$$\bar{U}_1(P, Q) = \int_0^1 U_1(v_1; P, Q) dF_1(v_1) \quad (12)$$

The probability of a type v_1 seller's trade is calculated as follows:

$$Q_1(v_1) = \int_0^1 Q(v_1, v_2) dF_2(v_2) \quad (13)$$

Buyers can benefit from similar formulas. We can precisely calculate a type v_i trader's interim expected utility from this equilibrium as $U_i(v_i; P', Q') = U_i(v_i; P, Q)$ since any combination of P' and Q' in the k -double auction implements an incentive-compatible double auction mechanism (P, Q) as in Eqs. 9 and 10. $\bar{U}_i(P', Q')$ and $Q_i(v_i; P', Q')$ can be calculated in a similar manner.

When for all $i \in 1, 2$ and all $v_i \in [0, 1]$, $U_i(v_i; P, Q) \geq 0$, a double auction mechanism (P, Q) is considered *individually rational*. In equilibrium, each trader's strategy implies that a loss never occurs; thus, the auction mechanisms outlined by k -double auction are considered individually rational [42]. Therefore, the proposed PB-DODAM is both incentive-compatible and individually rational, and the defined PB-DODAM mechanism is thus incentive feasible. \square

Receiving higher chances of trading for the agents who have a shorter trading time-frame is the main strategy of the criticality-based allocation mechanism. In other words, the criticality-based allocation mechanism considers a higher trading priority for the asks/bids that have a shorter trading period. This approach can be misused by strategic agents if they divide their current bids into smaller ones with shorter time-frames to increase their priority. Although these kinds of misbehaviours could be easily prevented by charging entry fees for bids and asks, it is assumed that agents do not use such strategies.

Realistically, not all receiving asks and bids by the auctioneer have the same level of importance or criticality. Therefore, there is a need to define a defer time to check if there is any higher priority ask/bid with greater urgency. Moreover, there could be new asks or bids that could be a better fit and bring a higher social surplus. Finding an appropriate defer time is a complex process due to the

following reasons. On the one hand, waiting for more incoming asks and bids that are in critical condition or of higher priority can enhance the probability of finding better matches. On the other hand, postponing the matching process could cause opportunity losses, and as a result, increases trade failures. In short, finding the right waiting time for matching asks and bids is a complex process, and Definition 10 tries to clarify this process by defining the *defer rate* parameter.

Definition 10 (Defer rate) : Eq. 14 represents the defer rate of agent i 's ask or bid at time-slot t within period p .

$$e_i^p(t) = \frac{t' - a_i^p}{d_i^p - a_i^p} \quad (14)$$

In Eq. 14, the denominator is agent i 's departure minus agent i 's arrival within period p which is constant. The numerator is t' minus agent i 's arrival, where t' is equal to the real time passage and increases every minute. Since the denominator is a constant value, as time passes and t' increases, the total fraction increases as well.

The defer rate definition adds a new constraint to our priority-based allocation algorithm by which every matchable ask and bid can be traded only if their defer times are greater than or equal to ε , which is the defer rate threshold. The functionality of ε is to provide enough waiting time to receive possible tasks with higher priorities and to find potentially better deals in our proposed mechanism. On the other hand, large values for the defer time threshold result in an increase in untraded matchable asks and bids which degrades the achieved performance and social welfare. This is why finding an appropriate ε is not a trivial task, and the value that we proposed for ε has come from an extensive number of multi-agent simulations experiments.

Algorithm 3 is the modified version of our priority-based allocation algorithm that considers the defer rate threshold as a new constraint to find proper matches.

Priority-based allocation with defer rate description

This modified version of the priority-based allocation algorithm provides the opportunity to defer the matching process to accept high-priority asks/bids that could join the system during the defer time. At the same time, choosing an appropriate threshold could potentially result in finding better matches. The first few steps of the current algorithm are similar to Algorithm 2. On line 5 of the current algorithm, it is verified that the sorted matchable bids collection is not null, and the defer rate of both asks and bids is greater than or equal to ε .

The goal of the modified mechanism is to provide enough waiting time to receive and accept potentially high-priority tasks. It also considers the departure time

Algorithm 3 Priority-based Allocation with Defer Rate

Input Stage: Prioritized asks and bids based on agents' satisfiability
Output Stage: VMs allocation based on prioritized asks and bids, considering ε threshold

- 1: Sort asks in descending order based on their updated priorities
- 2: **for** each seller i 's ask in time-slot t **do**
- 3: Find all matchable buyers' bids with seller i 's ask and update their priorities
- 4: Sort the machable bids based on their updated priorities in descending order
- 5: **if** (sorted machable bids \neq null) AND ($e_i^p(t) \geq \varepsilon$) AND ($e_j^p(t) \geq \varepsilon$) **then**
- 6: pair ask i and highest priority bid j
- 7: remove ask i and bid j from the sorted asks and bids lists
- 8: **else**
- 9: move ask i to the unsatisfied list and move on to the next ask
- 10: **end for**

to avoid trade failures. Avoiding trade failures ensures no drop in achieved resource allocation and overall utility. Adding the defer rate to the proposed basic algorithm brings our simulation environment remarkably closer to the real conditions of cloud ecosystems in which not all the tasks have equal priority. The enhancement distinguishes our research from similar state-of-the-art mechanisms.

Price-scheduling mechanisms

Providing an appropriate price-scheduling mechanism is an important key to success for economic models in cloud ecosystems. No double auction mechanism can satisfy all double auction properties, i.e. individual rationality, budget-balancing, truthfulness, and computational efficiency [42]. In most studies, appropriate price-scheduling mechanisms satisfy most of the above-mentioned aspects while keeping the rest at acceptable levels. price-scheduling mechanisms are important to avoid agents' strategic movements by taking the truthfulness into their consideration. This is a mandatory factor for market stability and agents' trade security. Moreover, pricing mechanisms can improve individual rationality, which motivates the agents to take part in the business based on their received benefits. Furthermore, individual rationality can increase the efficiency of the resource allocation by encouraging the participants to trade as many resources or services as possible. Our proposed pricing mechanism requires an increase in allocation efficiency and prevent strategic agents' behaviours to minimize trade failures. To guarantee that the auctioneer

does not take the short in position role, our proposed mechanism should be a strong balanced budget in which the auctioneer neither makes nor loses any benefit in the trades. The proposed pricing mechanism needs to fulfill the aforementioned features.

There is a number of price-scheduling mechanisms available in the market for double auction environments, such as Vickrey, McAfee, and k-double auction (k-DA) [18]. Among these price-scheduling mechanisms, k-DA guarantees the individual rationality, balanced budget, and truthfulness while providing a reasonable level of efficiency. As the k-DA's features are aligned with our proposed priority-based double auction mechanism, we have modified k-DA's price-scheduling system to be tailored to our needs.

k-DA's pricing mechanism is defined as follows:

Definition 11 (k-Double Auction (k-DA)) : *Considering m number of sellers and n number of buyers which are matchable, \bar{a}_m is the highest valuation among the matched asks and \underline{b}_n is the lowest valuation among the matched bids. The market price, $\phi(t)$, at time-slot t within period p is calculated according to Eq. 15:*

$$\phi(t) = (1 - k)\bar{a}_m + k\underline{b}_n \quad (15)$$

In Eq. 15, k is a variable between 0 and 1. It is proven that when the number of asks and bids increases, the k-DA pricing mechanism converges towards strategy-proofness [42]. Strategy-proofness is one of the main properties in DA truthfulness that provides the higher efficiency and lower trade failures, which are the main focus of the current paper.

In the competitive market of cloud ecosystems, service providers do not have motivation to exaggerate their actual valuation v_i^p to increase the market price. Increasing the market price risks the sellers' matching possibilities. On the other hand, underestimating their true valuation would decrease their benefits as our proposed allocation mechanism reduces the trade loss potentials, and thus it is needless to lower down the actual valuation. Due to the highly competitive cloud service markets, the distance between providers' asks and users' bids is quite small. As the users' tasks are often time-critical, there is no incentive to decrease the bidding valuation v_j^p as it can jeopardize their trades and miss their deadlines.

The way that agents propose their bids and reveal their types is a part of the bidding strategies. In any complex market such as cloud ecosystems, we can apply a limited number of bidding strategies to obtain experimental analyses. In the current research, we present a number of bidding strategies to simulate the various types of sellers and buyers in diverse market circumstances. To this end, the bidding strategies that are considered are built upon

common human characteristics in the face of priority-based services. A number of agents' bidding strategies are studied in the following part.

Agents' bidding strategies

The true valuation of seller i and buyer j are represented by v_i^p and v_j^p , respectively, while the reported valuation of seller i and buyer j are denoted by \hat{v}_i^p and \hat{v}_j^p , respectively. When the value of \hat{v}_i^p is considerably greater than the value of v_i^p and the value of \hat{v}_j^p is considerably lower than the value of v_j^p , the agents are highly motivated to increase the number of successful trades. If the values of \hat{v}_i^p are slightly different from the values of v_i^p , the agents only act for the trades that are the most beneficial to them, which is also known as aggressive bidding strategy. Aggressive bidding strategy inevitably increases the trade failure rates.

The followings are the most common bidding strategies in current markets:

(i) Modest strategy (MODS):

Modest strategy is a common bidding mechanism for perishable goods in spot markets, which technically converts double-sided auctions into one-sided auctions. The reported valuation of sellers in this approach would be always zero, as follows:

$$\hat{v}_i^t = 0.0 \quad (16)$$

In this strategy sellers do not participate in reporting their valuations and only buyers offer their bids.

(ii) Truthful strategy (TS):

In this strategy, both sellers and buyers report their actual asks and bids truthfully. In Eq. 17, seller i reports its valuation in time-slot t within period p as follows:

$$\hat{v}_i^t = v_i^p \quad (17)$$

In Eq. 18, buyer j reports its valuation in time-slot t within period p as follows:

$$\hat{v}_j^t = v_j^p \quad (18)$$

(iii) Monotonous strategy (MONOS):

In this strategy, sellers and buyers, after entering the auction at arrival time, adjust their valuation report over time, as well as considering the remaining time. Seller i 's valuation is reported at time-slot t within period p as follows:

$$\hat{v}_i^t = v_i^p (1.0 + \delta) \frac{d_i^p - t'}{d_i^p - a_i^p} \quad (19)$$

Buyer j 's valuation is reported at time-slot t within period p as follows:

$$\hat{v}_j^t = v_j^p \left(1.0 - \delta \frac{d_j^p - t'}{d_j^p - a_j^p} \right) \quad (20)$$

In Eqs. 19 and 20, we consider δ to be a parameter to manage the aggressiveness of the sellers' and the buyers' bidding attitude. When δ has a higher value, the agents are greedier for trading, whereas when δ has a lower value, the agents have more tendency for aggressive tradings. As we get closer to the departure time, the seller i 's reported valuation tends uniformly from $v_i^p (1.0 + \delta)$ to 0.0. Similarly, as we get closer to the departure time, the buyer j 's reported valuation tends uniformly from $v_j^p (1.0 - \delta)$ to v_j^p . This is one of the most common dynamic pricing strategies that are extensively used in the revenue management fields.

(iv) Aggressive strategy (AGS):

The AGS is technically a randomized version of the MONOS strategy. In AGS, when the trade failure risk is low, the agents attempt to achieve a higher social welfare, and when the offers get closer to the departure time, the agents ignore the profit and focus more on the successful trades. The seller i 's valuation is reported at time-slot t within period p as follows:

$$\hat{v}_i^t = \text{rand} \left(v_i^p (1.0 + \delta) \frac{d_i^p - t'}{d_i^p - a_i^p}, v_i^p (1.0 + \delta) \right) \quad (21)$$

The buyer j 's valuation is reported at time-slot t within period p as follows:

$$\hat{v}_j^t = \text{rand} \left(v_j^p (1.0 - \delta), v_j^p \left(1.0 - \delta \frac{d_j^p - t'}{d_j^p - a_j^p} \right) \right) \quad (22)$$

In Eqs. 21 and 22, a $\text{rand}(x, y)$ function is used which provides a random number between x and y . The seller i 's reported valuation starts with $v_i^p (1.0 + \delta)$ and ends with a random value within $[0.0, v_i^p (1.0 + \delta)]$ range. The buyer j 's reported valuation starts with $v_j^p (1.0 - \delta)$ and ends with a random value within $[v_j^p (1.0 - \delta), v_j^p]$ range.

These are the dominant bidding strategies that are widely used in the market and we have utilized the TS in our multi-agent simulation mechanisms to evaluate our proposed priority-based double auction mechanism.

With the proposed allocation approach, price-scheduling system and the agents' bidding strategies, we promote a priority-based online double auction mechanism that considers time-criticality of asks and bids and perishable nature of the offered services. As the proposed approach lowers the unsuccessful trades, it provides a higher resource allocation efficiency and a higher social welfare for the participants. Moreover, based on the modified k-DA price-scheduling mechanisms that is used for our proposed model, the auctioneer creates neither profit

nor loss that makes our proposed mechanism a strong balanced budget. The strong balanced budget prevents the short in position role for the auctioneer and increases the incentive compatibility of the system. Our proposed algorithms are evaluated using a multi-agent simulation method in the next section.

Experimental results

We have examined our model using a simulation environment to implement a number of scenarios in a repeatable controlled testbed. The experiments consume a reasonable amount of time and close-to-zero cost. Considering real IaaS secondary markets that will serve ground for the auction, we have tried our best to define a variety of conditions with realistic data. These conditions and realistic data allow the experimental results to represent the real outcomes.

Experiment setup

In our simulation environment, there are 10 service providers as sellers and 10 service users as buyers that participate in the market in 5 separate time-slots, and every time-slot lasts for 30 min. Some cloud providers on the market, such as Amazon, offer their services on the hourly basis. In their case, it makes no difference how many minutes of the hour does the user require. In our approach, we define the time-slot as the 30 min period. In fact, changing the length of this segment to any shorter or longer value does not impact the generality of the model. In every time-slot, sellers and buyers attend the auction in random arrival times offering their asks and bids, respectively, to find their match and start trading. These agents leave the auction after staying for a random number of minutes in the current time-slot, not exceeding the 30-min time-slot window. In our proposed model, cloud service providers in secondary markets supply VMs as the cloud trading units in which every cloud service user could demand a number of them. In every time-slot, a trade happens when a seller's ask and a buyer's bid satisfy the matching conditions. The matching conditions are satisfied when the quantity of VMs offered by a service provider is greater than or equal to a service user's demand, and the service provider's ask is less than or equal to the service user's bid. Moreover, an ask and a bid should have time overlap during their presence in the current time-slot to be able to trade. Service providers and service users can freely alter their asks' and bids' valuation according to their bidding strategies. At the end of each time-slot, the market is cleared, and there is no overlap between these time-slots.

In the classical price-based approach, pairing up the lower asks and higher bids is the only concern, and the priority and time-criticality of the tasks are taken for granted. In our proposed model, the time-criticality of

cloud offered services is of particular importance and forms the core component of our research. Based on overall condition of every ask and bid in each time-slot, we calculate the ask satisfiability and bid satisfiability, respectively. In every time-slot, if the demand for an ask is more than the total supplied VMs, the ask benefits from a higher satisfiability, compared to a situation when the demand is less than the total supplied VMs. In every time-slot, if the ask satisfiability is less than a certain threshold, it is considered that the ask is in a critical status and needs to be given a higher priority. Likewise, in every time-slot, if the supply for a bid is more than the total demanded VMs, the bid benefits from a higher satisfiability, compared to the situation when the supply is less than the total demanded VMs. In every time-slot, if the bid satisfiability is less than a certain threshold, it is considered that the bid is in a critical status and needs to be given a higher priority. This idea shapes the foundation of our proposed priority-based double auction mechanism to define whether each ask or bid has a critical condition or not.

Finding appropriate ask satisfiability threshold (AST) and bid satisfiability threshold (BST) is not a trivial task and is of particular importance which directly impacts our algorithm's performance. Based on a significant number of experiments, a range of AST and BST was acquired for our simulations. Defining a range of AST and BST is necessary to calculate the criticality and subsequently to assign the task priority. On the basis of the numerous experiments, it was concluded that the range of AST should be different from the range of BST. To investigate the impact of these thresholds on our proposed mechanism's performance, 0.0, 0.25, 0.5, 0.75 and 1.0 are tested as AST range and 0.0, 0.75, 1.5, 2.25 and 3.0 as BST range. To clarify the comparison between the existing dominant approach (the classical price-based mechanism) and our proposed method, we considered $AST=0.0$ and $BST=0.0$ for the classical priced-based mechanism, which in practice will ignore the time-criticality and perishability conditions. In every time-slot for each ask and bid, the satisfiability is calculated (Eqs. 5 and 7). Satisfiability threshold can be considered as the threshold of acceptable and normal conditions that any value less than that indicates critical conditions. In every minute in each time-slot if an ask satisfiability is less than the AST, we consider the calculated ask criticality (Eq. 6) as the ask priority. In this condition, the ask priority would increase when the time passes to increase the chances of conducting a successful trade. Likewise, in every minute in each time-slot, if a bid satisfiability is less than the BST, the calculated bid criticality (Eq. 8) becomes the bid priority. In this case, the bid priority would increase when the time passes to increase the chances of conducting a successful trade. Otherwise, if the ask satisfiability and bid satisfiability are not less than their designated thresholds, their conditions for the trade are

considered normal, and the price-based allocation priority will be used.

To recognize the amount of time that the current asks and bids can wait for any high-priority or urgent incoming task, we have defined a defer rate range containing 0.0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4 and 0.45. For instance, when the defer rate is equal to 0.05, agent i defers the matching process for 5% of their total presence in the current time-slot of the auction. By using this range, it can be understood how long we could wait for any high-priority task before facing any tangible drop in the matchability rate and the overall utility. The outcomes shown in the following sections are the averages of five hundred simulation runs for a low-risk trading market and the same number of runs for a high-risk trading market. Running the simulation this number of times ensures that the achieved results are stable and have a low variance.

A brief summary of simulation parameters is mentioned in Table 2. In the following subsection, we examine two possible scenarios in common IaaS secondary markets in which perishable or priority-based cloud resources or services are traded.

Description of scenarios

The low-risk trading market for PB-DODAM mechanism

In a low-risk trading market for our proposed mechanism, the average sellers' valuation is lower than the average

buyers' valuation. This difference increases the matching probability and decreases the risk of trade failure. In a low-risk trading market, the competition is normally moderate. In our experiments in a low-risk trading market, the sellers' valuation range is between 5 to 20 financial units, whereas the buyers' valuation is ranged between 10 to 30 financial units.

The high-risk trading market for PB-DODAM mechanism

In a high-risk trading market for our proposed mechanism, the average sellers' valuation is similar to the average buyers' valuation. The proximity of the sellers' and buyers' valuation negatively impacts the matching probability and increases the risk of failure. In a high-risk trading market, the competition is normally higher than in a low-risk trading market. In our experiments in a high-risk trading market, the sellers' and the buyers' valuation ranges are similar and between 10 to 30 financial units.

Figures 2 and 3 illustrate a low-risk trading market and a high-risk trading market, respectively. Figure 2 represents that the majority of asks and bids match, whereas in Fig. 3, almost half of the asks and bids could engage in trade successfully. In a low-risk trading market, the trading options experience less critical conditions, whereas in a high-risk trading market, when pairing options are diminished, we move towards more critical conditions.

The purpose of this paper is to investigate the effect of time-criticality (Eqs. 6 and 8) and task-priority factors on the overall matchability and social welfare. To this end, we need to examine the impact of different AST and BST on time-criticality and task priority.

In the following parts, we will explain the impact of different ranges of AST and BST in allocation mechanisms and social welfare evaluations. Moreover, in our research, we study the impact of different defer rates on our allocation and social welfare evaluation experiments. Our goal is to find out the amount of time that the current asks and bids can wait for any high-priority task without facing a tangible drop in the overall allocation and utility performance.

Allocation mechanism evaluation

Matching rate is defined as an index to indicate the number of successful trades. In Fig. 4, we illustrate the matching rate in a low-risk trading market scenario using different ranges for AST and BST. When the AST and the BST are equal to zero, no ask or bid is considered to be in a critical condition, and in this case, it falls into a price-based category. As it is depicted in the graph, the price-based mechanism (AST = 0 and BST = 0) has the lowest matching rate, whereas applying the least amount of thresholds (AST=0.25 and BST = 0.75) at once creates a leap in the matching rate. Increasing the satisfiability threshold for both asks and bids increases the matching

Table 2 Simulation parameters

Parameters	Value
Number of service providers (Sellers)	10
Number of service users (Users)	10
Number of time-slots (ts)	5
Number of minutes in each time-slot (t)	30
Total simulation time	150 min
Total number of experiments	500 times
Ask Satisfiability Threshold (AST)	0.0, 0.25, 0.5, 0.75, 1.0
Bid Satisfiability Threshold (BST)	0.0, 0.75, 1.5, 2.25, 3.0
Ask satisfiability	Calculated for each ts
Bid satisfiability	Calculated for each ts
Ask criticality	Calculated for every t
Bid criticality	Calculated for every t
Ask priority	Calculated for every t
Bid priority	Calculated for every t
Defer rate range	0.0 : 0.05 : 0.45
Low-risk seller valuation range	5 to 20
Low-risk buyer valuation range	10 to 30
High-risk seller valuation range	10 to 30
High-risk buyer valuation range	10 to 30

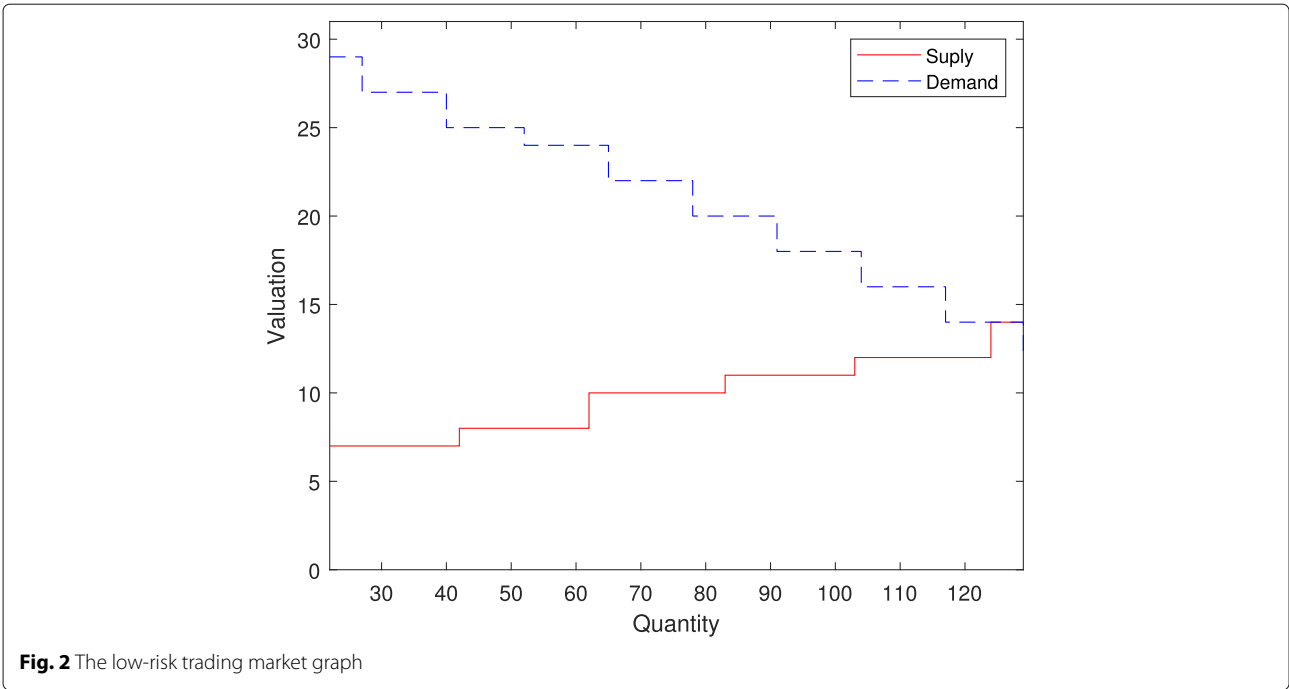


Fig. 2 The low-risk trading market graph

rate as well. As it is illustrated in Fig. 4, the matching rate values in the last two satisfiability thresholds are converged.

It can be analyzed that by defining a satisfiability threshold, the critical asks and bids which perish soon will receive higher priorities to trade. In this condition, the matching rate and the number of successful trades

increase, and this proves the proposed idea of the current paper.

By running an extensive number of simulations, it was found that in the last two satisfiability thresholds, the results were very close to each other. Increasing the satisfiability thresholds will not produce better results. A hundred of experiments were run to find out the best AST

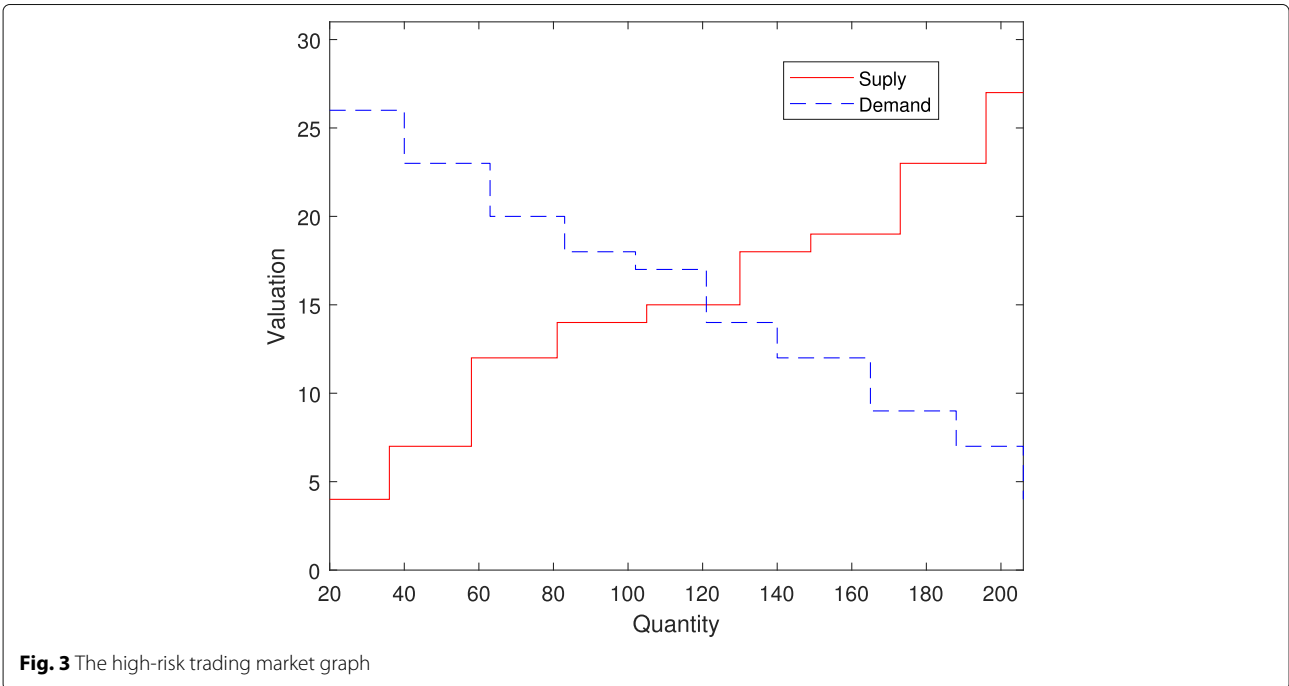


Fig. 3 The high-risk trading market graph

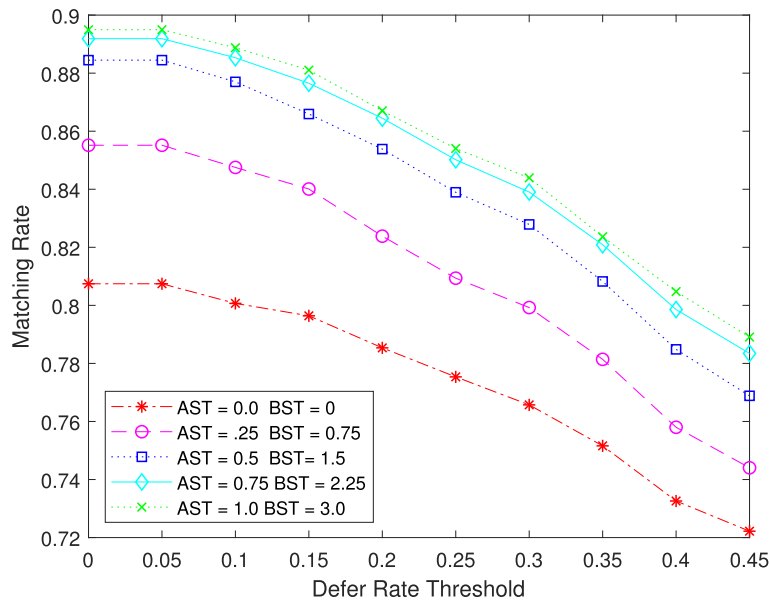


Fig. 4 Low-risk matching rate graph

and BST to achieve the highest matching rate. In short, 1.0 for the AST and 3.0 for the BST gained the highest matching rate among all tested values.

Another aspect to be considered in Fig. 4 is the impact of the defer rate on the matching rate. The defer rate determines how much the matching process can be delayed for any urgent or high-priority request without a noticeable drop in the performance. For all designated thresholds in the low-risk market scenario, the allocation

mechanism experiences no performance loss, when the defer rate is equal to 0.05, compared to the case when the defer rate is equal to zero. Moreover, when the defer rate is equal to 0.1, an intangible drop would happen in the matching performance, compared to the zero defer rate case.

Figure 5 illustrates the matching rate in a high-risk trading market using different ranges for the AST and the BST. Similar to Fig. 4, when the satisfiability thresholds

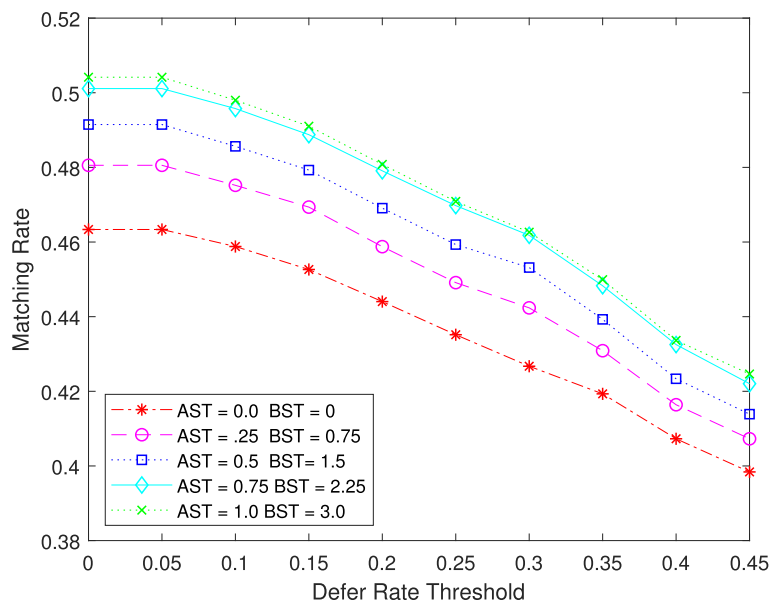


Fig. 5 High-risk matching rate graph

are equal to zero, the case falls into a category of price-based mechanisms, and applying the non-zero thresholds provides a noticeable difference that can be seen in the performance of the system. Moreover, like the low-risk trading market, the matching rate graphs for the last two thresholds converge. From the defer rate perspective, if the defer rate is equal to 0.05, there is no drop in the matching rate efficiency, compared to the case that the defer rate is equal to zero. In this case, the system can defer the trading process for high-priority tasks without any performance loss.

A comparison of the low-risk and the high-risk trading markets' graphs shows that regardless of the threshold factor, the overall success rate of trading in the high-risk markets is considerably lower than the low-risk markets. The reason for that is in the high-risk markets, the asks' and the bids' valuations are close to each other. In this case, the competition is high, and so are the chances of the trade failures. Regardless of the AST and the BST ranges and the defer rate range, in terms of matching rate, the low-risk trading markets perform tremendously better than the high-risk ones. In terms of the defer rate, both low-risk and high-risk trading markets have the same performance in the range of 0 and 0.05, and no drop is experienced in them. Both low-risk and high-risk markets have relatively similar behaviour when the defer rate is between 0.05 to 0.1, while the low-risk market shows a slightly better performance in terms of the matching rate. The reason for this is when the competition is low, the matching process can be deferred for slightly longer, compared to the case when the competition is high.

In addition to the matching rate increment, which is the main purpose of the current paper, it is expected that the number of successful trades increases, and consequently, the overall social welfare improves. In the following section, the social welfare will be evaluated based on the sellers, the buyers and the overall social welfare in both low-risk and high-risk markets. The illustrated graphs are the result of averaging five hundred times of simulation runs for the low-risk and the high-risk market scenarios.

Social welfare evaluation

Social welfare evaluation is one of the most common metrics to evaluate the profitability of a system. It can be inferred that there is a direct relationship between the resource allocation and the financial outcome of a system. First, we examine the social welfare in a low-risk trading market to measure the financial aspects of our proposed mechanism. For this purpose, we will examine the performance of the sellers, the buyers, and the total utility separately. We use the k-DA price-scheduling mechanism and the TS bidding strategy to calculate the utility of the sellers, the buyers, and the overall social welfare.

Figures 6 and 7 illustrate the sellers' utility and the buyers' utility performance in a low-risk trading market. When the AST and the BST are equal to zero, we experience the price-based mechanism which has the lowest utility for both sellers and buyers. Applying the least amount of thresholds makes a significant difference in the efficiency, as illustrated in Figs. 6 and 7. In these figures, the two highest thresholds result in the maximum amount of benefits to sellers and buyers. The achieved utility of the

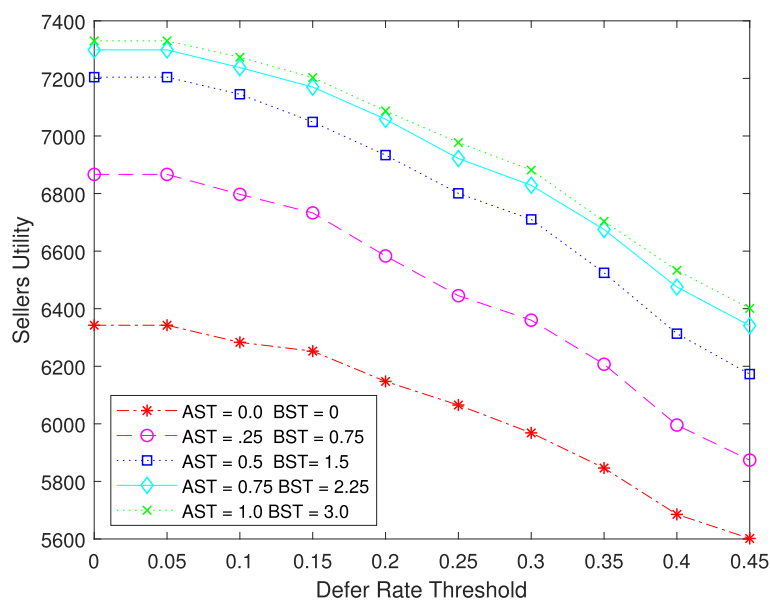


Fig. 6 Low-risk sellers' utility graph

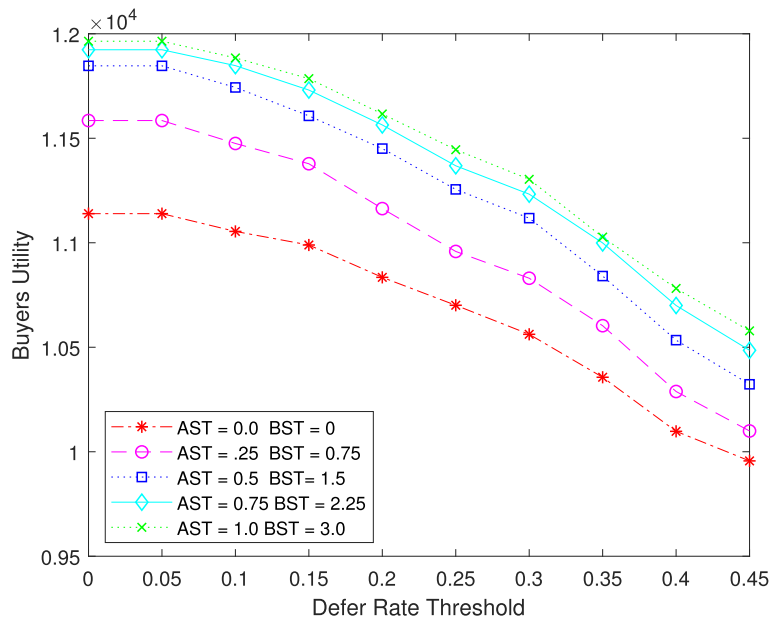


Fig. 7 Low-risk buyers' utility graph

last two thresholds are close to each other, and any further increment will bring no more benefits to the engaged agents. If we consider the time-criticality of the current asks and bids and prioritize the asks and bids that are close to expiration, we will increase the trade chances. This can lead to more successful trades and consequently brings more profit to participating agents, as illustrated in Figs. 6 and 7.

Figure 8 is the summation of all participating agents' utilities in a low-risk trading market and shows how well our mechanism works in terms of profitability by applying appropriate ASTs and BSTs.

In the following paragraphs, we examine the social welfare in a high-risk trading market to measure the sellers, the buyers and the overall social welfare in separate scenarios.

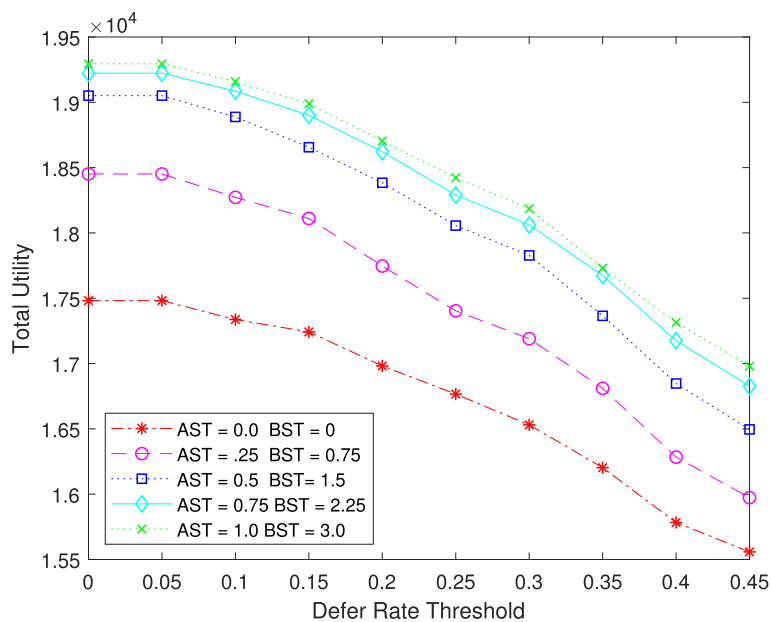


Fig. 8 Low-risk total utility graph

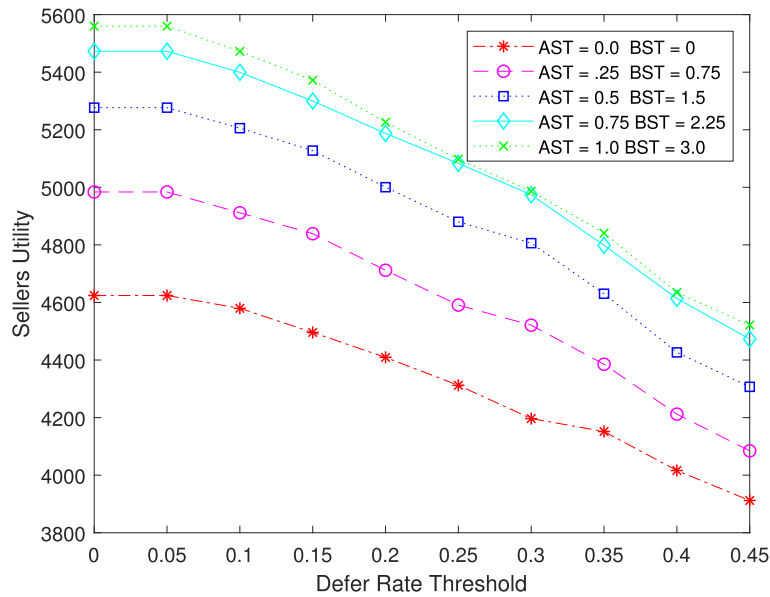


Fig. 9 High-risk sellers' utility graph

Figures 9 and 10 illustrate the sellers' and the buyers' utility performance in a high-risk market. Applying the minimum AST and BST to distinguish normal cases and critical ones drastically improves sellers' and buyers' utility, compared to the price-based mechanism where the AST and the BST are equal to zero. The two maximum AST and BST results are close to each other and bring the highest social welfare to the participating agents.

Figure 11 is the summation of all participating agents' utilities in a high-risk trading market and shows how well our mechanism works in terms of financial outcome by applying appropriate ASTs and BSTs.

In summary, in low-risk trading markets, the sellers' valuations are relatively lower than the buyers' valuations. Hence, the competition and consequently, the chances of trade failures are relatively low. Unlike low-risk trading markets, in high-risk markets, the sellers' and the buyers'

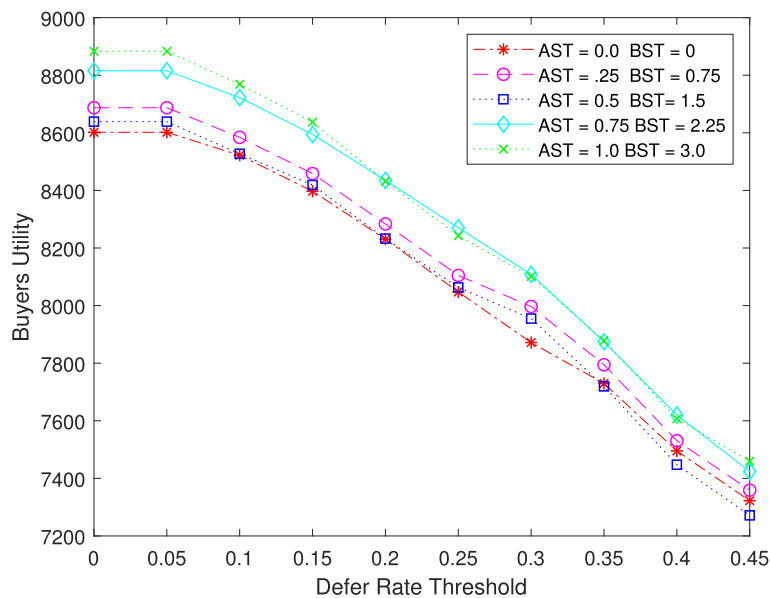


Fig. 10 High-risk buyers' utility graph

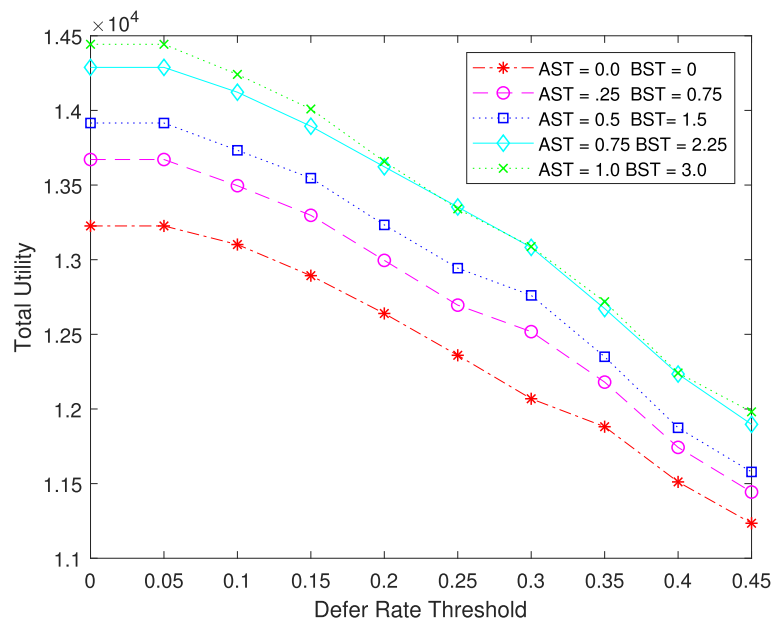


Fig. 11 High-risk total utility graph

valuations are close to each other, and as a result, the competition and the chances of trade failure are comparatively high. This leads to lower social welfare, compared to the low-risk trading markets.

Regardless of the trading market types, the more critical tasks are considered and given higher priority, the more successful trades and social welfare are achieved. Based on the illustrated results, it can be asserted that in both low-risk and high-risk trading markets, considering the AST and the BST to prioritize the critical tasks drastically increases the matching rate and the overall social welfare of the system.

Conclusions

In conventional double auction mechanisms, the priority and the time constraints of cloud tasks are not addressed well. This causes some resources to perish and lost due to the time constraints in the IaaS secondary markets. In convectional double auction mechanisms, the priority and time constraints of cloud tasks are not addressed well. This causes some resources to be perished and lost due to time constraints. In this paper, by considering the perishability and time-criticality of the cloud resources and services in the IaaS secondary markets, a priority-based dynamic online double auction mechanism was proposed. In this mechanism, the time-criticality and the availability of cloud resources and requests determined the matching priority for the agents' asks and bids. We investigated the validity of our proposed mechanism in both low-risk and high-risk trading market scenarios by conducting a tremendous number of simulation runs. In this regard,

it was found that setting appropriate thresholds for ask satisfiability and bid satisfiability increases the trading chances for critical tasks. Considering time-critical tasks increases the matching rate and accordingly improves the overall social welfare. In our proposed model, the performance of the system in the allocation and social welfare domains are significantly improved over the classical price-based approach. Moreover, we defined a defer rate to explore the amount of time that the matching process can be delayed to accommodate any incoming high-priority request before facing a tangible drop in the matchability and the overall utility.

In this research, a k-double auction price-scheduling mechanism was applied, and future work can explore the other price-scheduling models to enhance our current mechanism. Moreover, the truthful strategy was the main bidding strategy in the current research, and future work can consider the impact of applying the other bidding strategies. The current paper proposed a multi-unit priority-based double auction mechanism for cloud ecosystems, and in future work, priority-based combinatorial double auction models can explore more detailed cloud components. Moreover, future studies could investigate different priority classifications to enhance the current priority-based double auction mechanism.

Abbreviations

PB-DODAM: Cloud priority-based dynamic online double auction mechanism; SaaS: Software as a service; PaaS: Platform as a service; IaaS: Infrastructure as a service; CA-LP: Combinatorial auction-linear programming; CA-GREEDY: Combinatorial auction-greedy; CDARA: Combinatorial double auction resource allocation; FMCDAM: Fair multi-attribute combinatorial double auction model; QoS: Quality of service; DCA: Double-sided combinatorial

auction; TC: Transparent computing; INP: Infrastructure providers; SP: Service providers; VM: Virtual machine; TMDA: Truthful multi-unit double auction model; VCG: Vickrey-Clarke-Groves; PUE: Power usage effectiveness; MBM: Maximum-weighted bipartite matching; IISG: Imperfect information Stackelberg game; S: Seller; B: Buyer; k-DA: k-double auction; MODS: Modest strategy; TS: Truthful strategy; MONOS: Monotonous strategy; AGS: Aggressive strategy; AST: Ask satisfiability threshold; BST: Bid satisfiability threshold

Acknowledgments

The authors gratefully acknowledge the contributions of Nadia Mokhireva and Chloe Grove for their professional help on the original version of this document that greatly assisted the research.

About the authors

S. M. Reza Dibaj

Reza holds a Bachelor's degree in Computer Software Engineering from the Tehran Azad University (2000), Iran. He holds a Master's degree in Information Technology - Computer Networks from Tehran Polytechnic University (2012), Iran, and he is doing his Ph.D. in Computer Science in Ryerson University, Canada. His research interests include Cloud Computing, Distributed Systems, Energy Efficiency, and Data Science. He has more than 15 years full-time experience as a College Instructor, Senior Expert IT Developer, and System Analyst.

Ali Miri

Ali Miri has been a Full Professor at the School of Computer Science, Ryerson University, Toronto. He has over 25 years of research experience in security and privacy technologies and their applications, computer networks and digital communication, and cloud computing and big data. He has authored and co-authored over 220 refereed manuscripts, including 6 books, and 8 patents in these fields. Dr. Miri has served on more than 100 organizing and technical program committees of international conferences and workshops, and has been the main organizer of over a dozen international conferences. He has supervised over 90 students, and has also overseen the successful completion of a large number of industry-related security projects, which have received close to \$7M in funding. He has served as an editor for a number of international journals. He regularly acts as a consultant to industry on various projects, and is on a number of steering or advisory boards. He is a senior member of the IEEE, and a member of the Professional Engineers of Ontario.

SeyedAkbar Mostafavi

Seyed Akbar Mostafavi received his B.S degree in Information Technology from Sharif University of Technology (SUT), Tehran, Iran, in 2008. He received his M.S. degree in computer networks from Amirkabir University of Technology (AUT), Tehran, Iran in 2010, and his Ph.D. degree in computer networks from Amirkabir University of Technology (AUT), Tehran, Iran in 2014. He is an active consultant in the field of information technology enterprise architecture and computer networks and has conducted several research projects in this field. He has served as the reviewer for several computer journals and as a TPC member for several computer conferences. His current research area includes machine learning, cloud computing and the Internet of Things (IoT).

Authors' contributions

All authors have equally contributed to the preparation of this manuscript.

Funding

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

Availability of data and materials

The data is generated by a multi-unit simulation environment and is available upon request.

Competing interests

There are no financial or non-financial competing interests among the author regarding this publication.

Author details

¹Department of Computer Science, Ryerson University, Toronto, Canada.

²Department of Computer Engineering, Yazd University, Yazd, Iran.

Received: 29 November 2019 Accepted: 4 November 2020

Published online: 23 November 2020

References

- Dibaj SMR, Sharifi L, Miri A, Zhou J, Aram A (2018) Cloud computing energy efficiency and fair pricing mechanisms for smart cities. In: 2018 IEEE Electrical Power and Energy Conference (EPEC). IEEE. pp 1–6. <https://doi.org/10.1109/epec.2018.8598406>
- Weinman J (2011) Time is money: the value of "on-demand". JoeWeinman.com, Jan 7:30
- JoSEP AD, Katz R, KonWinSKI A, Gunho L, PAttERson D, RABKin A (2010) A view of cloud computing. Commun ACM 53(4):50–58
- Varian HR, Farrell J, Shapiro SC (2004) The Economics of Information Technology: An Introduction. Cambridge University Press, Published in the United States of America by Cambridge University Press
- Patel CD, Shah AJ (2005) Cost model for planning, development and operation of a data center. Hewlett-Packard Lab Tech Rep 107:1–36
- Barroso LA, Hölzle U (2007) The case for energy-proportional computing. Computer 40(12):33–37
- Sandholm T, Ortíz JA, Odeberg J, Lai K (2006) Market-based resource allocation using price prediction in a high performance computing grid for scientific applications. In: 2006 15th IEEE International Conference on High Performance Distributed Computing. IEEE. pp 132–143. <https://doi.org/10.1109/hpdc.2006.1652144>
- Stage A, Setzer T (2009) Network-aware migration control and scheduling of differentiated virtual machine workloads. In: 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing. IEEE. pp 9–14. <https://doi.org/10.1109/cloud.2009.5071527>
- Clearwater SH, Huberman BA (2005) Swing options: a mechanism for pricing it peak demand. In: 11th International Conference on Computing in Economics and Finance. Society for Computational Economics, Washington DC
- Rogers O, Cliff D (2012) A financial brokerage model for cloud computing. J Cloud Comput Adv Syst Appl 1(1):1–12
- Cartlidge J, Clamp P (2014) Correcting a financial brokerage model for cloud computing: closing the window of opportunity for commercialisation. J Cloud Comput 3(1):2
- Shi W, Zhang L, Wu C, Li Z, Lau F (2014) An online auction framework for dynamic resource provisioning in cloud computing. In: ACM SIGMETRICS Performance Evaluation Review, vol. 42. ACM. pp 71–83. <https://doi.org/10.1145/2591971.2591980>
- Samimi P, Teimouri Y, Mukhtar M (2016) A combinatorial double auction resource allocation model in cloud computing. Inform Sci 357:201–216
- Toosi AN, Vanmechelen K, Khodadadi F, Buyya R (2016) An auction mechanism for cloud spot markets. ACM Trans Auton Adapt Syst (TAAS) 11(1):2
- Talluri KT, Ryzin GV (2004) The Theory and Practice of Revenue Management. 1st edn, Vol. 68. Springer, New York
- Kong X, Huang GQ, Luo H, Yen BP (2018) Physical-internet-enabled auction logistics in perishable supply chain trading: State-of-the-art and research opportunities. Ind Manag Data Syst 118(8):1671–1694
- Cheng M, Xu SX, Huang GQ (2016) Truthful multi-unit multi-attribute double auctions for perishable supply chain trading. Transp Res Part E Logist Transp Rev 93:21–37
- Miyashita K (2014) Online double auction mechanism for perishable goods. Electron Commer Res Appl 13(5):355–367
- Singh S, Chana I (2016) A survey on resource scheduling in cloud computing: Issues and challenges. J Grid Comput 14(2):217–264
- Kong W, Lei Y, Ma J (2016) Virtual machine resource scheduling algorithm for cloud computing based on auction mechanism. Optik 127(12):5099–5104
- Nejad MM, Mashayekhy L, Grosu D (2014) Truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds. IEEE Trans Parallel Distrib Syst 26(2):594–603
- Kumar D, Baranwal G, Raza Z, Vidyarthi DP (2017) A systematic study of double auction mechanisms in cloud computing. J Syst Softw 125:234–255
- Wang H, Kang Z, Wang L (2015) Performance-aware cloud resource allocation via fitness-enabled auction. IEEE Trans Parallel Distrib Syst 27(4):1160–1173

24. Patel YS, Nighojkar A, Misra R (2019) Truthful double auction based vm allocation for revenue-energy trade-off in cloud data centers. In: 2019 National Conference on Communications (NCC). IEEE. pp 1–6. <https://doi.org/10.1109/ncc.2019.8732201>
25. Wei W, Fan X, Song H, Fan X, Yang J (2016) Imperfect information dynamic stackelberg game based resource allocation using hidden markov for cloud computing. *IEEE Trans Serv Comput* 11(1):78–89
26. Anjum N, Karamshuk D, Shikh-Bahaei M, Sastry N (2017) Survey on peer-assisted content delivery networks. *Comput Netw* 116:79–95
27. Mostafavi S, Dehghan M (2016) Game-theoretic auction design for bandwidth sharing in helper-assisted p2p streaming. *Int J Commun Syst* 29(6):1057–1072
28. Lu L, Yu J, Zhu Y, Li M (2018) A double auction mechanism to bridge users' task requirements and providers' resources in two-sided cloud markets. *IEEE Trans Parallel Distrib Syst* 29(4):720–733
29. Zhao D, Zhang D, Perrussel L (2011) Mechanism design for double auctions with temporal constraints. In: Twenty-Second International Joint Conference on Artificial Intelligence. AAAI Press, Palo Alto
30. Zaman S, Grosu D (2013) Combinatorial auction-based allocation of virtual machine instances in clouds. *J Parallel Distrib Comput* 73(4):495–508
31. Baranwal G, Vidyarthi DP (2015) A fair multi-attribute combinatorial double auction model for resource allocation in cloud computing. *J Syst Softw* 108:60–76
32. Pla A, Lopez B, Murillo J (2015) Multi-dimensional fairness for auction-based resource allocation. *Knowledge-based Syst* 73:134–148
33. Chen Y, Zhang Q (2015) *Dynamic Spectrum Auction in Wireless Communication*. 1st edn. Springer, New York
34. Prasad GV, Prasad AS, Rao S (2016) A combinatorial auction mechanism for multiple resource procurement in cloud computing. *IEEE Trans Cloud Comput* 6(4):904–914
35. Bahreini T, Badri H, Grosu D (2018) An envy-free auction mechanism for resource allocation in edge computing systems. In: 2018 IEEE/ACM Symposium on Edge Computing (SEC). IEEE. pp 313–322. <https://doi.org/10.1109/sec.2018.00030>
36. Wang J, Liu A, Yan T, Zeng Z (2018) A resource allocation model based on double-sided combinational auctions for transparent computing. *Peer-to-Peer Netw Appl* 11(4):679–696
37. Li Q, Huang C, Bao H, Fu B, Jia X (2019) A game-based combinatorial double auction model for cloud resource allocation. In: 2019 28th International Conference on Computer Communication and Networks (ICCCN). IEEE. pp 1–8. <https://doi.org/10.1109/iccn.2019.8846922>
38. Dibaj SMR, Miri A, Mostafavi S (2020) A cloud dynamic online double auction mechanism (DODAM) for sustainable pricing. *Telecommun Syst*. <https://doi.org/10.1007/s11235-020-00688-4>. in press, The article is available as 'Online First': <http://link.springer.com/article/10.1007/s11235-020-00688-4>
39. Yang D, Zhang X, Xue G (2014) Promise: A framework for truthful and profit maximizing spectrum double auctions. In: IEEE INFOCOM 2014-IEEE Conference on Computer Communications. IEEE. pp 109–117. <https://doi.org/10.1109/infocom.2014.6847930>
40. Green J, Laffont J-J (1979) *Incentives in Public Decision-making*, vol. 1. Elsevier North-Holland, New York
41. Myerson R, Satterthwaite M (1983) Efficient Mechanism Design for Bilateral Trading. *J Econ Theory* 28:265–81
42. Satterthwaite MA, Williams SR (1989) Bilateral trade with the sealed bid k-double auction: Existence and efficiency. *J Econ Theory* 48(1):107–133

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
