RESEARCH

Open Access

A truthful dynamic combinatorial double auction model for cloud resource allocation



Qihui Li^{1*}, Xiaohua Jia^{2,3} and Chuanhe Huang²

Abstract

Dynamic auction-based resource allocation models require little global price information, are decentralized and suitable for the distributed systems like cloud computing. For the cloud computing market, we proposed a Truthful Dynamic Combinatorial Double Auction (TDCDA) model to improve the social welfare and resource utilization. In our model, multiple cloud service providers and cloud users bid for various resources in a dynamic environment. We adopted a payment scheme to ensure truthfulness for all participants, which motivates bidders to reveal their true preferences. Since the combinatorial auction allocation with goal of economic efficiency is NP-hard, we developed a greedy mechanism to achieve the approximately efficient solution. Considering both parties' interests and the resource scarcity, this model also ensures fairness and balances resource allocation. The proposed model is proven to be approximately efficient, incentive compatible, individually rational and budget-balanced. Simulation results show that the model not only achieves economic efficiency, but also improves resource allocation and meets resource needs for more cloud users.

Keywords Cloud computing, Combinatorial double auction, Bid density, Truthfulness, Economic efficiency, Resource allocation, Winner determination problem (WDP)

Introduction

As a new business model, cloud computing uses internet and virtualization to provide ready-to-use IT resources. It provides efficient and scalable computing resources, reduces setup and maintenance costs, and improves the resource utilization [1].

With the help of virtualization technology, cloud computing integrates a large number of IT resources (including storage, CPUs, networks, applications, etc.), and provides them to users based on supply and demand. It is convenient for users and has many advantages over

liqihui@hgnu.edu.cn

³ Department of Computer Science, City University of Hong Kong, Hong Kong, China

the traditional model. However, it also brings some challenges to resource management [2].

Users' requirements could change at any time and cloud resources are heterogeneous and expandable. Therefore, the cloud market changes constantly and is highly dynamic. When demand increases, it is difficult to ensure Quality of Service (QoS). How to schedule and allocate cloud resources to make an optimal match between supply and demand is one of the major challenges in cloud computing [3].

The competition for cloud resources in cloud computing is similar to that in the free market in economics. Many models and theories in economics have been applied to cloud computing. Current researches usually use the auction method from economics to simulate cloud resource allocation in order to optimize the social welfare. However, the resources in cloud computing are diverse and cloud users' requirements are different. The allocation problem of combinatorial cloud resources is NP-Hard [4].



© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

^{*}Correspondence:

Qihui Li

¹ Department of Computer Science, Huanggang Normal University, Huanggang, China

² School of Computer Science, Wuhan University, Wuhan, China

The fixed price strategy adopted in the traditional auction method does not truly reflect the service qualities and request differences, and there are deviations in fairness and resource utilization. Cloud service providers need to maximize their profits, and cloud users want to spend less and obtain resources with desired quality. They have their own interests, which often conflict with each other. A transaction can only be reached when both parties' interests are satisfied. Therefore, when designing a cloud resource allocation mechanism, we need to take both parties' interests into consideration to build a healthy and sustainable cloud computing market.

Game theory studies the interaction of multiple players in a competitive environment [5]. It can be used to solve the optimization problem with competition or mutual constraints among multiple players in economics, international relations, biology, computer science, etc. In the cloud computing market, both cloud service providers and cloud users are driven by their own interests, and there are competition and mutual constraints. Game theory can be used to meet the needs of both parties.

Two popular categories of market-based models for cloud resource management are commodity market model and auction model. In the commodity market model, providers specify their resource prices and charge users according to the amount of resources they would consume. In the auction model, each provider and user acts independently and individually. Auctions are used for products that have no standard values and the prices are affected by supply and demand at a specific time [6]. Auctions require little global price information, are decentralized, easy to implement and suitable for distributed systems like grid computing, cloud computing, etc.

Auction is a type of game that bidders need to strategically select their best bids. It is also one of the many ways to implement the dynamic pricing. Dynamic pricing reflects the real-time supply and demand relationship, one example is the Amazon EC2 spot market. It is desirable for the dynamic cloud market as trade prices can be adjusted based on the supply and demand changes to obtain the efficient allocation. Among the different types of auctions, combinatorial double auction is most suitable for cloud computing because it allows double-sided competition and bidding on any combination of resources, which helps to achieve the economic efficiency. An effective auction mechanism needs to have the desirable properties of economic efficiency, incentive compatibility/truthfulness, individual rationality, budget-balance and computational efficiency.

Motivation and contributions

Since there is competition between cloud service providers and cloud users, they will not disclose all of their information, so the auction is an incomplete information game [7]. The bidders are self-interested and intend to maximize their own utilities. Some bidders might benefit by manipulating the market with untruthful bids and gain unfair advantages. That could hinder other qualified bidders and reduce the auction efficiency. Most researches considered truthfulness for either the cloud users or cloud service providers. We adopted a payment scheme to ensure truthfulness for all participants. In our model, bidding truthfully (revealing true information) is the dominant strategy for every participant. As the combinatorial auction allocation with goal of economic efficiency is NP-hard, we developed a greedy mechanism to mitigate the computational complexity of combinatorial auctions. Experimental results show that our algorithm can balance resource allocation, and achieve optimal social welfare (sum of utilities for all participants) and high request success rate.

The main contributions of the paper are as follows:

- This paper proposed a Truthful Dynamic Combinatorial Double Auction (TDCDA) model to ensure truthfulness for all participants.
- In the proposed model, we developed a greedy mechanism to overcome the complexity of computing the efficient allocation in combinatorial auctions.
- Considering the resource scarcity, the proposed model balances resource allocation for cloud service providers.

In the double-sided auction, both cloud users and cloud service providers can submit bids, and their utilities are included in the winner determination problem (WDP). All participants are treated equally, the proposed model is fair enough to motivate bidders to participate and stay in the market. The TDCDA model is proven to be approximately efficient, incentive compatible, individually rational, budget-balanced and computationally efficient.

Related work

Cloud computing integrates a large amount of IT resources through virtualization technology and provides scalable resources on demand. It improves resource utilization, and reduces operation and maintenance costs. However, the dynamically changing resources and requests complicate cloud resource allocation and management. In recent years, resource allocation is a major challenge for the cloud market. How to effectively allocate these huge and dynamic resources has been an important issue that needs to be solved. Some theories and models in economics are very suitable for the cloud resource allocation problem. Researchers have used the game theory and auction methods to study the cloud resource allocation process.

Game theory studies strategic interaction between rational decision-makers [7]. Liu et al. [8] studied ad hoc cloud providers selling their private idle computing resources to cloud users through non-cooperative game to improve resource utilization and reduce maintenance costs. Jie et al. [9] modeled the resource allocation problem as a double-stage Stackelberg game and proposed three algorithms to achieve Nash equilibrium and Stackelberg equilibrium. Lin et al. [10] proposed a strategic game approach to address the multi-ownership of the network resources and services, and analyzed the Nash equilibrium of the game and stability of the algorithm. Zheng et al. [11] have investigated the problem of multiuser computation offloading for mobile cloud computing under dynamic environment. By formulating this problem as a stochastic game, they proved the dynamic offloading decision process always leads to a pure-strategy Nash Equilibrium. Chen et al. [12] proposed a stochastic scheduling algorithm for the cloud computing environment to optimize the accumulative QoS. Guo et al. [13] modeled the bandwidth sharing problem as a Nash bargaining game, and proposed the allocation principles by defining a tunable base bandwidth for each virtual machine (VM).

Auction is designed with certain bidding strategies and rules, where sales can be done with bids. Researchers have conducted extensive study on auction-based resource allocation schemes. Zhang et al. [14] designed an efficient and truthful online auction for dynamic resource scaling and pricing. Li et al. [15] adopted a truthful and individual-rational double auction as the inter-cloud trading mechanism, and designed a dynamic algorithm for each cloud to decide the best VM valuation and bidding strategies. Zhang et al. [16] proposed an incentive-Compatible Online Cloud Auction (COCA) mechanism based on a monotonic payment rule and a utility-maximizing allocation rule. Tafsiri et al. [17] studied combinatorial double auction-based market in which a broker performs the allocation of the providers' virtual machines according to the users' requests. Their proposed integer linear programming model achieved truthfulness, fairness and allocation efficiency. Mashayekhy et al. [18] proposed optimal and approximate strategyproof mechanisms for resource management in clouds with multiple types of physical machines and resources that give incentives to the users to reveal their true valuations for the requested bundles of virtual machines. Li et al. [19] proposed a game-based combinatorial double auction model for resource allocation supporting multiple infrastructure providers and service providers bidding for various combinations of resources. Considering both parties' interests, this model optimized social welfare and achieved high resource utilization. Dibaj et al. [20] proposed a cloud priority-based dynamic online double auction mechanism, which is aligned with the dynamic nature of cloud supply and demand and the agents' time constraints. Umer et al. [21] proposed Adaptive Market-Oriented Combinatorial Double Auction Resource Allocation (AMO-CDARA) model that allocates services to users based on multiple parameters such as less price, QoS, and ranking of providers. It reduced service-level agreement (SLA) violation, improved users' satisfaction and the social welfare. Ullah et al. [22] proposed a Negotiation based Combinatorial Double Auction mechanism for Resource Allocation (N-CDARA) in cloud computing. The proposed method negotiates with dropped users to maximize the number of winners. It also improved the resource utilization and utilities. Lee et al. [23] presented two energy-conscious task consolidation heuristics to maximize resource utilization and improve energy efficiency. Yadav et al. [24] proposed three adaptive models to minimize energy consumption and SLA violation.

Compared with other auction mechanisms, combinatorial double auction not only solves the problem of monopoly but also reduces transaction times and costs. Many researchers have studied combinatorial double auction for cloud computing. An auction mechanism basically consists of two parts, i.e., allocation function and payment scheme, which need to be designed carefully to achieve the auction properties [25]. However, it is impossible to have all the auction properties, like computational efficiency and economic efficiency conflict with each other in combinatorial double auction. Therefore, researchers have focused on the trade-off and proposed various mechanisms that satisfy a subset of those properties [26].

The combinatorial auction allocation with goal of economic efficiency is often denoted as the WDP that is NP-hard. In order to solve the WDP, Sandholm et al. [27] developed a sophisticated search algorithm— Branch On Bids (BOB) including structural improvements, optimizations at search nodes, and faster data structures. Sandholm et al. [28] designed a depth-first branch-and-bound search algorithm—Combinatorial Auction Branch On Bids (CABOB), and it has better anytime performance than CPLEX. Lehmann et al. [29] studied combinatorial auctions with single-minded bidders and proposed an \sqrt{M} greedy scheme which approximated the optimal allocation, where M is the total number of resources. Zaman et al. [30] extended the greedy mechanism and redefined M to be the weighted total number of VM instances. They proposed a Combinatorial Auction-Greedy (CA-GREEDY) mechanism and showed that CA-GREEDY can allocate VM instances efficiently. Nejad et al. [31] designed a family of truthful greedy mechanisms considering heterogeneity and the scarcity of cloud resources. However, they only considered bids from users. Samimi et al. [32] extended mechanism in [30] and defined bid density for both cloud service providers and cloud users.

In auctions, incentive compatibility/truthfulness is one of the most critical properties. It is essential to resist market manipulation and ensure auction fairness and efficiency. Researchers have designed different pricing schemes to ensure truthfulness. Vickrey auction [33] is an example of truthful auction mechanism. Vickrey-Clarke-Groves (VCG) mechanism is strategy-proof and efficient. However, it suffers from computational complexity. Li et al. [34] proposed combinatorial double auction for the resource allocation and pricing in grid system, and calculated trade price as average of the bid prices of the matching provider and user. Samimi et al. [32] extended models in [30] and [34], and proposed a Combinatorial Double Auction Resource Allocation (CDARA) model in cloud computing. Although their mechanisms claimed to be incentive compatible through the experimental studies, the average price is not an incentive compatible mechanism theoretically. Di et al. [35] extended the traditional second-price bidding policy to a novel double-sided nextprice bidding policy, and ensured truthfulness for both consumers and providers. It is also approved that their resource allocation scheme is ex-post efficient. Kumar et al. [36] proposed a Truthful Multi-unit Double Auction (TMDA) mechanism and calculated the trade price following the procedure of dominant strategy double auction in McAfee [37].

So far, there are not many researches on truthfulness for all participants. Most of the reported work ensures truthfulness for either the cloud users or cloud service providers. Kumar et al. [36] considered truthfulness for all participants with multi-unit resources, but they did not consider multiple types of resources. We considered the heterogeneity in resources, and proposed a truthful dynamic combinatorial double auction model to ensure truthfulness for all participants. To mitigate the computational complexity of combinatorial auctions, we developed a greedy approximation method to solve the winner determination problem. The proposed model is proven to be approximately efficient, incentive compatible, individually rational, budget-balanced and computationally efficient.

Although there are lots of researches on combinatorial double auction for cloud computing, it is rarely implemented in the real cloud market. There are still many challenges need to be resolved such as perishable resources, QoS of the allocated resources, SLA violation, energy consumption, cloud users' satisfaction, providers' reputation, data security, bidder drop problem, etc.

Truthful Dynamic Combinatorial Double Auction (TDCDA) model

The cloud resources are provided as three major types of services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). IaaS provides CPUs, storage, networks and other infrastructure resources; PaaS provides a platform for software developers to build their applications; and SaaS provides already created applications. Our targeted system is a generic cloud computing environment.

For the auction-based lease model of cloud market, the market framework is composed of multiple cloud users (CUs), multiple cloud service providers (CSPs) and the auctioneer. In the leasing process, various cloud resources need to be arranged differently based on cloud users' requirements.

Model construction

In the combinatorial double auction, both CUs and CSPs submit their bids to the auctioneer who determine the winners and trade prices. Bidders' valuations for the resources are their private information. We set up a model as shown in Fig. 1 to realize their optimal utilities.

Cloud service provider

Cloud service providers offer resources and act as sellers. They submit bids with their available resources and ask prices. For cloud service providers, the resource costs, energy consumption, resource quantities and QoS are different. Therefore, different cloud service providers have different costs for the resources.

Cloud user

Cloud users intend to purchase resources and act as buyers. They submit bids with their resource requirements and bid prices. For cloud users, they have different time and resource requirements, different price acceptance ranges, etc., so their bids are different.



Fig. 1 Combinatorial double auction model

Auctioneer

Auctioneer is in charge of the auction and acts as an interface between CSPs and CUs. It collects bids from all participants, and uses the allocation algorithm to obtain winners and trade prices.

Model assumptions

For the combinatorial double auction model, we made the following assumptions:

- 1. Bidders in the auction are rational and do not collude, their goals are to maximize their respective utilities.
- 2. Bidders' valuations of the resources are their private information. They do not know other participants' valuations.
- 3. Cloud users are single-minded and bid only one specific set of resources. They will pay only when they receive the entire set.
- 4. A cloud user can only trade with one cloud service provider.

Our model assumes that the auctions run periodically and cloud users request only one set of resources in each round. The assumption of single-minded users does not limit them to express more flexible requirements and they can revise bids based on their preferences. For example, a cloud user might have some changes that need to purchase some other types of resources, then it could update bid in the next round.

Model analysis

There are N cloud resource providers, M cloud users, and k kinds of resources in the auction. Figure 2 shows the sequence diagram for the proposed model.

- 1. Begin
- 2. Auctioneer starts auction
- 3. CUs and CSPs submit bids
- 4. Auctioneer acknowledges all participants' bids
- 5. Auctioneer closes the auction
- 6. Auctioneer informs all participants about the auction close



Fig. 2 Sequence diagram of the TDCDA model

- 7. Based on the proposed model, auctioneer obtains the winners and trade prices
- 8. Auctioneer sends the auction result to all participants
- 9. Winning CSPs allocate resources to winning CUs based on the auction result
- 10. CUs pay according to the final trade prices
- 11. End

In combinatorial auctions, participants can bid for combinations of items rather than just individual items. This allows bidders to express their preferences more fully which could improve the economic efficiency, especially when items are complements [38]. Our model allows cloud users to place bids on bundles of resources to express complementarity. For cloud service providers, as they know their costs for each type of resources, their bids include per-unit ask prices.

Cloud service provider *i*'s resource capacity vector is $C_i^S = (c_{i1}^S, ..., c_{ir}^S, ..., c_{ik}^S), B_i^S = (b_{i1}^S, ..., b_{ir}^S, ..., b_{ik}^S)$ is its unit ask price vector. c_{ir}^S and b_{ir}^S are cloud service provider *i*'s capacity and unit ask price of the *r*-th resource, respectively.

Cloud user *j*'s resource quantity vector (bundle) is $C_j^U = \left(c_{j1}^U, ..., c_{jr}^U, ..., c_{jk}^U\right)$. b_j is its bid price for the bundle. c_{jr}^U is cloud user *j*'s requested quantity of the *r*-th resource.

Each participant's benefit in a combinatorial auction is expressed by its utility. Under normal circumstances, when the cloud service provider could fulfill the cloud user's request and the cloud user's bid price is no less than the cloud service provider's ask price, a deal can be reached.

When cloud service provider *i* and cloud user *j* reach a deal, cloud user *j* pays p_{ij}^{U} and cloud service provider *i* receives p_{ij}^{S} . Cloud service provider *i*'s utility can be represented as the difference between the payment it received and its valuation; and cloud user *j*'s utility can be calculated as the difference between its valuation and the price it paid [37]. In our truthful model, all participants would bid for resources according to their true valuations and we get the utility functions for cloud service provider *i* and cloud user *j* as given in Eqs. (1) and (2), respectively:

$$u_{ij}^{S} = p_{ij}^{S} - \sum_{r=1}^{k} b_{ir}^{S} c_{jr}^{U}$$
(1)

$$u_{ij}^{U} = b_j - p_{ij}^{U} \tag{2}$$

Our model adopts dynamic pricing strategies, it improves resource utilization and solves the unfairness problem of fixed price in the traditional model. Also, it allocates combinatorial resources at one time, which is more efficient than multiple allocations for each kind of resources.

Winner determination

Given the information received from all participants, the model determines the winners in the auction system in such a way that the social welfare (sum of utilities for all participants) get maximized. The combinatorial auction problem is denoted as the winner determination problem. We considered utilities for cloud service providers, cloud users and the auctioneer. Auctioneer's utility is included as the model could not operate without the auctioneer [37].

When cloud service provider *i* and cloud user *j* reach a deal, cloud user *j* pays p_{ij}^{U} , cloud service provider *i* receives p_{ij}^{S} , and the auctioneer's utility is $p_{ij}^{U} - p_{ij}^{S}$. $s_{ij} = \sum_{r=1}^{k} b_{ir}^{S} c_{jr}^{U}$ is cloud service provider *i*'s total ask price for cloud user *j*'s request. The allocation problem is formulated as follows:

$$max \left(\sum_{j=1}^{M} \sum_{i=1}^{N} \left(p_{ij}^{S} - s_{ij} \right) x_{ij} + \sum_{j=1}^{M} \sum_{i=1}^{N} \left(b_{j} - p_{ij}^{U} \right) x_{ij} + \sum_{j=1}^{M} \sum_{i=1}^{N} \left(p_{ij}^{U} - p_{ij}^{S} \right) x_{ij} \right) \\ = max \sum_{j=1}^{M} \sum_{i=1}^{N} \left(b_{j} - s_{ij} \right) x_{ij}$$
(3)

The first item inside the big brackets is the sum of utilities obtained by all cloud service providers, the second item is the sum of utilities obtained by all cloud users, and the third item is utility obtained by the auctioneer. They can be simplified to be the right side of the equation, and we get the WDP as:

$$max \sum_{j=1}^{M} \sum_{i=1}^{N} (b_j - s_{ij}) x_{ij}$$
(4)

$$s.t. \sum_{j=1}^{M} c_{jr}^{U} x_{ij} \le c_{ir}^{S}$$

$$\tag{5}$$

$$\sum_{i=1}^{N} x_{ij} \le 1 \tag{6}$$

$$\begin{aligned} x_{ij} &\in \{0, 1\} \\ 0 &< i \leq N \\ 0 &< j \leq M \\ 0 &< r \leq k \end{aligned}$$
 (7)

Constraint (5) indicates that the total amount of the rth resource requested by all cloud users that traded with cloud service provider i does not exceed the amount of the *r*-th resource owned by cloud service provider *i*. Constraint (6) indicates that one cloud user cannot trade with more than one cloud service provider. Constraint (7) indicates that when cloud service provider *i* and cloud user *j* reach a deal, x_{ij} is 1; otherwise, it is 0. It is noted that the social welfare does not depend on the final trade prices, but the prices do affect bidders' utilities.

The combinatorial double auction allocation problem is NP-hard. We extended mechanisms in [31, 32], and defined bid density for cloud service providers and cloud users considering the scarcity of resources.

Cloud user j's bid density D_j is:

$$D_j = \frac{b_j}{\sqrt{\sum_{r=1}^k f_r c_{jr}^U}}$$
(8)

Cloud service provider *i*'s bid density D_i is:

$$D_{i} = \frac{\sum_{r=1}^{k} b_{ir}^{S} c_{ir}^{S}}{\sqrt{\sum_{r=1}^{k} f_{r} c_{ir}^{S}}}$$
(9)

Where f_r is:

$$f_r = \frac{1}{\sum_{i=1}^{N} c_{ir}^S}$$
(10)

We defined f_r as the scarcity of the *r*-th resource, which depends on the capacity of the *r*-th resource.

Based on their bid density, the cloud users and cloud service providers will be sorted in descending and ascending order, respectively. The main reason for sorting is to prioritize the higher valued cloud users and lower valued cloud service providers, and then find the most profitable trades by matching the cloud users whose bids are higher with the cloud service providers whose asks are cheaper in turn.

When a cloud user requests a scarce resource, its bid density will be lower which means its request would be less likely to be fulfilled. On the other hand, if a cloud service provider has the scarce resource, its bid density will also be lower. As the cloud service providers are ordered in ascending order of their bid density, it will make the cloud service provider more popular. In this way, we can balance the resource allocation for cloud service providers.

Payment determination

After sorting cloud users and cloud service providers based on their bid density, we tried to find potential trades in turn. When cloud service provider *i* could fulfill cloud user *j*'s request and $s_{ij} \leq b_j$, they could reach a deal and we will calculate their average prices.

Average price for cloud user *j*'s bid:

$$a_j^U = \frac{b_j}{\sum_{r=1}^k c_{jr}^U} \tag{11}$$

Average price for cloud service provider *i*'s ask based on cloud user *j*'s resource request:

$$a_{ij}^{S} = \frac{s_{ij}}{\sum_{r=1}^{k} c_{jr}^{U}}$$
(12)

Suppose there are l pairs of trade candidates, we sort the average prices in descending order for cloud users as given in (13) and ascending order for cloud service providers as given in (14), respectively.

$$a_1^U \ge a_2^U \dots \ge a_{j'}^U \dots \ge a_l^U \tag{13}$$

$$a_1^S \le a_2^S \dots \le a_{i'}^S \dots \le a_l^S \tag{14}$$

Compare these average prices, and we might find $a_1^U \geq a_1^S$, or

$$a_R^U \ge a_T^S \ge a_{R+1}^U \tag{15}$$

And

$$a_{T+1}^S \ge a_R^U \ge a_T^S \tag{16}$$

If $a_l^U \ge a_l^S$, then all the trade candidates with indices less than l can be completed, cloud users pay price $p_{ij}^{U} = a_l^{U} \sum_{r=1}^{k} c_{jr}^{U} \text{ and cloud service providers receive}$ $payment <math>p_{ij}^{S} = a_l^{S} \sum_{r=1}^{k} c_{jr}^{U}.$ Otherwise, define $a_0 = (a_{T+1}^{S} + a_{R+1}^{U})/2$. If $a_0 \in [a_T^{S}, a_R^{U}]$,

the trade candidates with indices $j' \leq R$ and $i' \leq T$

can complete transactions, and the trade price is can complete transactions, and the trade price is $p_{ij}^{U} = p_{ij}^{S} = a_0 \sum_{r=1}^{k} c_{jr}^{U}$. If $a_0 \notin [a_T^S, a_R^U]$, the trade candidates with indices j' < R and i' < T can complete transactions, cloud users pay price $p_{ij}^{U} = a_R^U \sum_{r=1}^{k} c_{jr}^U$ and cloud service providers receive payment $p_{ij}^S = a_T^S \sum_{r=1}^k c_{jr}^U.$

Figure 3 shows how to get indices R and T. Sometimes it might not be possible to satisfy both (15) and (16). For example, if T = l, then we can only get (15). We assumed $a_{l+1}^{\hat{S}} = \infty$ and $a_{l+1}^{U} = 0$. In this way, indices R and T can always be found to satisfy both (15) and (16).

In order to ensure truthfulness for all participants, some least profitable trades might not be able to complete with this pricing mechanism. However, as the auction runs periodically, the lost trades might be completed in the following rounds.

Auction properties

An effective market-based resource allocation model must meet the needs of both cloud service providers and cloud users, and encourage them to participate in the market. Our proposed model was assessed from an economic perspective, and the following properties are fulfilled:

Economic efficiency

An efficient mechanism maximizes the total social welfare. As the combinatorial auction allocation problem with goal of economic efficiency is NP-hard, we adopted a greedy mechanism which is approximately efficient [29-32].



Fig. 3 Average prices for potential trading CUs and CSPs

Incentive compatibility

A double auction mechanism is incentive compatible/ truthful if bidding truthfully is the dominant strategy for every participant. Truthfulness is essential to resist market manipulation and ensure efficiency. In untruthful auctions, some bidders can manipulate their bids to obtain outcomes that favor themselves but hurt others. In truthful auctions, the resources can be allocated efficiently to users who value them the most. Our payment scheme ensures truthfulness for all participants, and it can be proved with the usual Vickrey argument.

Let us check about truthfulness for cloud service providers. In the proposed model, winning cloud service providers receive payment equal to the lowest average price of non-traded cloud service providers multiply the resource quantity, or $a_0 \sum_{r=1}^{k} c_{jr}^U$. Suppose a cloud service provider attempted to obtain better utility by misrepresenting its preference, the results can be classified as follows:

If bid value > true valuation, then the cloud service provider might be:

- still one of the winners, and there is no change in its utility.
- rejected by the model as its bid is too high and its utility becomes 0. If the cloud service provider could be a winner with true valuation, it suffers with the bid.

If bid value < true valuation, then the cloud service provider might be:

- still one of the winners, and there is no change in its utility.
- still one of the losers as its bid is too high and its utility remains 0.
- a winner from a loser. However, its utility will be negative and it suffers loss.

Therefore, truthful bidding is the dominant strategy for cloud service providers. Similarly, we can prove truthfulness for cloud users.

Budget-balance

Budget-balance property means that the total payment from all cloud users is no less than the total money received by all cloud service providers. This property ensures the viability and sustainability of the market by ensuring that the auctioneer is not at loss in conducting the auction. Our mechanism is budget-balanced as $a_R^U \ge a_T^S$ or $a_l^U \ge a_l^S$.

Individual rationality

A mechanism is said to be individually rational if each participant's utility is always non-negative. In our model, cloud users' payments are not greater than their bid prices, and cloud service providers receive no less than their ask prices. The trade price will be zero if no participant wins in the auction. Therefore, every participant gets either positive or zero utility, and our model is individually rational.

Computational efficiency

The greedy mechanism has high computational efficiency, as the resource allocation and payment can be decided in polynomial time [29]. The time complexity of TDCDA is O(kMN), which is polynomial.

Cloud resource allocation algorithm

This section explains the resource allocation algorithm. Most of researches just consider truthfulness for either cloud users or cloud service providers. This paper proposed a truthful dynamic combinatorial double auction (TDCDA) algorithm to ensure truthfulness for all participants.

Some researchers used average price for the combinatorial double auction. However, the average price is not a truthful mechanism theoretically. For example, a cloud service provider might try to ask price higher than its valuation, then it could benefit from higher trade price when the bid wins. Similarly, there are incentives for cloud users to try bid prices less than their valuations. If some participants do not reveal their true preferences, the most profitable trades might not occur between higher valued cloud users and lower valued cloud service providers. It will affect the social welfare and cause fairness problem.

Cloud resource allocation algorithm

Each cloud service provider submits its unit ask prices and resource quantities to the auctioneer. Each cloud user submits its bid price and requested quantities to the auctioneer. After receiving information from all parties, the auctioneer uses cloud resource allocation algorithm to obtain the allocation result that can maximize utilities for all participants. antities and bid prices submitted by all CUs

Input: The resource quantities and unit ask prices submitted by all CSPs, the resource

```
Output:
         x: trade relationship matrix
         p: trade price matrix, p_{ii} = (p_{ii}^U, p_{ii}^S)
1. Initialize: x = 0, p = 0:
2. CSPs and CUs submit bids to the auctioneer:
3. Calculate CUs' bid density D<sub>i</sub>:
4. Calculate CSPs' bid density D<sub>i</sub>;
5. Sort D_i such that D_1 \ge \dots \ge D_i \ge \dots \ge D_M;
6. Sort D_i such that D_1 \leq \ldots \geq D_i \leq \ldots \leq D_N;
7. while there is active cloud user do
      for j=1 to M do
8
9
         for i=1 to N do
           Calculate s_{ii} = \sum_{r=1}^{k} c_{ir}^{U} b_{ir}^{S};
10.
11.
            if b_i \ge s_{ii} and CSP i could fulfill CU j's request then
             Calculate CU j's and CSP i's average prices a_i^U and a_{ij}^S based on Eqs.(11) and (12),
12.
     respectively:
13.
             break:
            end if
14.
           end for
15
       end for
16.
17. Sort a_i^U such that \{a_1^U \ge \dots \ge a_i^U \ge \dots \ge a_i^U\}:
18. Sort a_{ij}^S such that \{a_1^S \leq \ldots \leq a_{i'}^S \leq \ldots \leq a_l^S\};
19. if a_i^U \ge a_i^S then
         Set x_{ii} = 1 for potential trades with indices less than l_i p_{ii}^S = a_i^S \sum_{\alpha=1}^k c_{i\alpha}^U and p_{ii}^U = a_i^S \sum_{\alpha=1}^k c_{i\alpha}^U
20.
     a_{l}^{U} \sum_{r=1}^{k} c_{ir}^{U};
21. else
         Find indices R and T to satisfy a_R^U \ge a_T^S \ge a_{R+1}^U and a_{T+1}^S \ge a_R^U \ge a_T^S;
22
23
         Set a_0 = (a_{T+1}^S + a_{R+1}^U)/2;
         if a_T^S \leq a_0 \leq a_R^U then
24
25.
          Set x_{ii} = 1 for potential trades with indices i' \leq R and i' \leq T;
26.
          Set p_{ij}^{S} = p_{ij}^{U} = a_0 \sum_{r=1}^{k} c_{jr}^{U};
27.
        else
28.
          Set x_{ii} = 1 for potential trades with indices j' < R and i' < T;
29.
           Set p_{ij}^{S} = a_{T}^{S} \sum_{r=1}^{k} c_{jr}^{U}, p_{ij}^{U} = a_{R}^{U} \sum_{r=1}^{k} c_{jr}^{U};
30.
         end if
31.
        end if
32.
        if x == 0 then
33.
        break: %no trade pair and exit
34. end if
35. Calculate new resource quantities for CUs and CSPs:
36. Calculate CUs' new bid density and sort them in descending order:
37. Calculate CSPs' new bid density and sort them in ascending order:
38 end while
39. Send x and p to all participants:
```

Algorithm. Cloud resource allocation

After receiving bids from all participants, auctioneer calculates the bid density for cloud users and cloud service providers, and arranges them in descending and ascending order, respectively. Then, it matches them one by one, if the cloud user's bid price is not less than the cloud service provider's ask price and its resource request could be satisfied by the provider, they could potentially trade. Then the auctioneer calculates their average prices based on Eqs. (11) and (12), and find the final payments and trade pairs. After that, it sends the auction result to all participants.

Simulation results

We implemented a simulation environment for cloud resource allocation using MATLAB, and compared our Truthful Dynamic Combinatorial Double Auction (TDCDA) algorithm with the Average price Combinatorial Double Auction (ACDA) algorithm. Cloud resource allocations for different numbers of cloud service providers (CSPs) and cloud users (CUs) were simulated. The bid prices of CUs and CSPs were generated randomly between the lowest and highest prices of the cloud resources. Bid quantities of resources for all participants were also generated randomly. We have completed multiple experiments, and compared results of the two algorithms for social welfare, resource allocation rate for cloud service providers, CUs' and CSPs' success rates.

Average price is often used for combinatorial double auctions. We compared our algorithm with the average price algorithm. Based on bid density, the two algorithms sort CUs and CSPs in descending and ascending order, respectively. Then TDCDA algorithm calculated the trade prices according to our truthful payment scheme. And ACDA algorithm calculated the trade prices as the average of bid prices of the matching CSPs and CUs.

Social welfare analysis

The social welfare obtained after competitive auctions with different numbers of CSPs and CUs are shown in Fig. 4. Figure 4a, b, c, and d show social welfare when the numbers of CSPs were fixed at 20, 40, 60 and 80 respectively, and the number of CUs increased by 20 from 20 and gradually reached 120.

We can see that when the number of CSPs and CUs is small, the social welfare obtained by the two algorithms are small and the difference is small. With the increasing number of CSPs and CUs, the social welfare obtained by the two algorithms increased, and the difference between the two algorithms also shows an increasing trend. Comparing with ACDA algorithm, the TDCDA algorithm increased social welfare by 30%.

As ACDA algorithm is not incentive compatible, some participants tried to benefit themselves by not bidding their true valuations. However, that could hurt other participants and reduce the auction efficiency. Some most profitable trades did not occur and the social welfare was affected. The problem became more obvious when the number of participants increased.

Resource allocation rate analysis

In resource allocation rate experiment, the number of CUs in each set of experiments was fixed. The resource



Fig. 4 Social welfare analysis

allocation rates for different numbers of CSPs are shown in Fig. 5. We can see that the same CSP obtained different resource allocation rates with the two algorithms.

There are 30 CSPs and 100 CUs in Fig. 5a. The horizontal axis represents the CSP sequence in ascending order of their bid density, and the vertical axis is resource allocation rate for the corresponding CSP.

We can see that the resource allocation rate of each CSP with TDCDA algorithm is less than 90%, and it shows a gradual downward trend as the bid density increases. Therefore, the TDCDA algorithm can balance resource allocation as we considered the resource scarcity.

With ACDA algorithm, we find that resource allocation rate fluctuated and some CSPs failed to trade even though they could trade if every participant bid truthfully as with TDCDA algorithm. Meanwhile, some CSPs have allocation rate more than 90%, which is likely to have overload issue. From the result, we can see that when there are participants bidding untruthfully, some CSPs lost unnecessarily and it is not beneficial for the whole market.

Figure 5b shows the allocation result for 60 CSPs and 100 CUs. Compared with Fig. 5a, TDCDA algorithm can still balance resource allocation between CSPs. However, we see more fluctuations and CSPs failed to trade with ACDA algorithm.

Request success rate analysis

In request success rate experiment, the number of CSPs in each set of experiments was fixed. We checked the ratio of the number of successfully allocated CUs to the total number of CUs that requested resources. Figure 6 shows the request success rate with the two algorithms when the number of CUs increases. In Fig. 6a and b, the numbers of CSPs were fixed at 20 and 80 respectively,



Fig. 6 Request success rate analysis

and the number of CUs increased by 20 from 20 and gradually reached 120.

We can see that, as the number of CUs increases, the request success rates obtained by the two algorithms show a downward trend. The TDCDA algorithm always has better success rate up to 20% higher than ACDA algorithm. This result indicates that TDCDA algorithm can satisfy a larger number of CUs.

With ACDA algorithm, some participants could benefit by not revealing their true preferences. With goal to maximize their respective utilities, they tried strategies best to themselves. However, that could hurt others and cause fairness issue. It discouraged some qualified bidders to participate and stay in the auction, and the request success rate was affected. TDCDA algorithm is truthful and unbiased for all participants. It is fair enough to motivate bidders to participate and stay in the auction. Their true bids also helped to achieve good request success rate.

Provider success rate analysis

In provider success rate experiment, the number of CUs in each set of experiments was fixed. We checked the ratio of the number of successfully traded CSPs to the total number of CSPs. Figure 7 shows the provider success rate with the two algorithms when the number of CSPs increases. In Fig. 7a and b, the numbers of CUs were fixed at 40 and 80 respectively, and the number of CSPs increased by 20 from 20 and gradually reached 120.



Fig. 7 Provider success rate analysis

As the number of CSPs increases, the provider success rates obtained by the two algorithms show a downward trend. The TDCDA algorithm always has better success rate than ACDA algorithm. In the resource allocation rate experiment, we find that some CSPs lost unnecessarily with ACDA algorithm. Figure 7 shows that TDCDA algorithm could improve the providers' success rate by 50%. This result indicates that more CSPs could trade with TDCDA algorithm.

In summary, the proposed TDCDA algorithm is superior to ACDA algorithm in terms of social welfare, resource allocation rate, request and provider success rates.

Conclusion

For the multi-tenant cloud computing market, this paper proposed a truthful dynamic combinatorial double auction allocation model. Since there is the competition between cloud service providers and cloud users, they will not share all of their information, so we adopted a payment scheme to ensure truthfulness for all participants. We also defined bid density for both cloud service providers and cloud users, and developed a greedy mechanism to obtain the approximately optimal allocations. Considering both parties' interests and the resource scarcity, this model also ensures fairness and balances resource allocation. Simulations show that compared with the average price combinatorial double auction algorithm, the proposed algorithm could increase social welfare by 30%, balance resource allocation, improve providers' success rate by 50% and meet resource needs for 20% more cloud users.



It is also proven that the proposed TDCDA model is approximately efficient, incentive compatible, individually rational, budget-balanced and computationally efficient.

There are still lots of challenges in cloud computing. Future work could include QoS, SLA violation, providers' reputation, users' satisfaction, energy consumption, etc.

Abbreviations

Truthful Dynamic Combinatorial Double Auction
Average price Combinatorial Double Auction
Cloud Service Provider
Cloud User
Winner Determination Problem
Quality of Service
Virtual Machine
Service-Level Agreement
Incentive-Compatible Online Cloud Auction
Adaptive Market-Oriented Combinatorial Double Auction
Resource Allocation
Negotiation based Combinatorial Double Auction mechanism
for Resource Allocation
Branch On Bids
Combinatorial Auction Branch On Bids
Combinatorial Auction-Greedy
Vickrey-Clarke-Groves
Combinatorial Double Auction Resource Allocation
Truthful Multi-unit Double Auction
Infrastructure as a Service
Platform as a Service
Software as a Service

Acknowledgements

The authors would like to thank the editors and reviewers who helped to improve the paper.

Authors' contributions

All authors have participated in conception and design, drafting the article and revising it critically for important intellectual content. All authors read and approved the final manuscript.

Authors' information

Qihui Li received the M.S. degree from Wuhan University of Technology in 2009, and the Ph.D. degree from Wuhan University in 2023. She is currently working at Huanggang Normal University. Her research interests include cloud computing, network functions virtualization and wireless networks. Xiaohua Jia (IEEE Fellow) received the B.Sc. and M.Sc. degrees from the University of Science and Technology of China in 1984 and 1986, respectively, and the D.Sc. degree in information science from The University of Tokyo in 1991. He is currently a Chair Professor with the Department of Computer Science, City University of Hong Kong. His research interests include cloud computing and distributed systems, data security and privacy, computer networks, and mobile computing. He is the General Chair of ACM MobiHoc 2008, a TPC Co-Chair of IEEE GLOBECOM 2010 Ad Hoc and Sensor Networking Symposium, and an Area Chair of IEEE INFOCOM 2010 and 2015-2017. He is an Editor of IEEE Internet of Things (2012–2018), IEEE Transactions on Parallel and Distributed Systems (2006–2009), Journal of Networks, Journal of World Wide Web, Journal of Combinatorial Optimization, etc.

Chuanhe Huang received his B.S., M.S., and Ph.D. degrees in Computer Science from Wuhan University in 1985, 1988, and 2002, respectively. He is currently a professor and Ph.D. supervisor at the School of Computer Science, Wuhan University. His research interests include SDN, opportunistic networks, and wireless networks, focusing on cryptography, wireless security, and trust management.

Funding

This work is supported by the National Natural Science Foundation of China (No.61772385).

Availability of data and materials

Not applicable.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 2 August 2021 Accepted: 26 June 2023 Published online: 18 July 2023

References

- Dillon T, Wu C, Chang E (2010) Cloud computing: issues and challenges. In: 24th IEEE International Conference on Advanced Information Networking and Applications. p 27–33. https://doi.org/10.1109/AINA.2010.187
- Galante G, de Bona L (2012) A survey on cloud computing elasticity. In: IEEE Fifth International Conference on Utility and Cloud Computing, p 263–270. https://doi.org/10.1109/UCC.2012.30
- Prasad A, Rao S (2014) A mechanism design approach to resource procurement in cloud computing. IEEE Trans Comput 63:17–30. https://doi. org/10.1109/TC.2013.106
- Gu S, Li Z, Wu C, Huang C (2016) An efficient auction mechanism for service chains in the NFV market. In: IEEE INFOCOM 2016 The 35th Annual IEEE International Conference on Computer Communications, p 1–9. https://doi.org/10.1109/INFOCOM.2016.7524438
- 5. Zhang W (2005) Game Theory and Information Economics. Shanghai People's Publishing House, Shanghai
- Izakian H, Abraham A, Ladani BT (2010) An auction method for resource allocation in computational grids. Futur Gener Comput Syst 26(2):228– 235. https://doi.org/10.1016/j.future.2009.08.010
- 7. Shen Q (2010) A course in game theory. China Renmin University Press, Beijing
- Liu G, Xiao Z, Tan G, Li K, Chronopoulos A (2020) Game theory-based optimization of distributed idle computing resources in cloud environments. Theoret Comput Sci 806:468–488. https://doi.org/10.1016/j.tcs.2019.08.019
- Jie Y, Guo C, Choo K, Liu C, Li M (2020) Game-theoretic resource allocation for fog-based industrial internet of things environment. IEEE Internet Things J 7:3041–3052. https://doi.org/10.1109/JIOT.2020.2964590
- 10. Lin T, Alpcan T, Hinton K (2017) A game-theoretic analysis of energy efficiency and performance for cloud computing in communication

networks. IEEE Syst J 11:649–660. https://doi.org/10.1109/JSYST.2015. 2451195

- Zheng J, Cai Y, Wu Y, Shen X (2019) Dynamic computation offloading for mobile cloud computing: a stochastic game-theoretic approach. IEEE Trans Mob Comput 18:771–786. https://doi.org/10.1109/TMC.2018.2847337
- Chen Y, Wang L, Chen X, Ranjan R, Zomaya A, Zhou Y, Hu S (2020) Stochastic workload scheduling for uncoordinated datacenter clouds with multiple QoS constraints. IEEE Trans Cloud Comput 8:1284–1295. https:// doi.org/10.1109/TCC.2016.2586048
- Guo J, Liu F, Zeng D, Lui J, Jin H (2013) A cooperative game based allocation for sharing data center networks. In: 2013 Proceedings IEEE INFOCOM, p 2139–2147. https://doi.org/10.1109/INFCOM.2013.6567016
- Zhang X, Huang Z, Wu C, Li Z, Lau F (2021) Dynamic VM scaling: Provisioning and pricing through an online auction. IEEE Trans Cloud Comput 9:131–144. https://doi.org/10.1109/TCC.2018.2840999
- Li H, Wu C, Li Z, Lau F (2013) Profit-maximizing virtual machine trading in a federation of selfish clouds. In: 2013 Proceedings IEEE INFOCOM, p 25–29. https://doi.org/10.1109/INFCOM.2013.6566728
- Zhang H, Jiang H, Li B, Liu F, Vasilakos A, Liu J (2016) A framework for truthful online auctions in cloud computing with heterogeneous user demands. IEEE Trans Comput 65:805–818. https://doi.org/10.1109/TC. 2015.2435784
- Tafsiri S, Yousefi S (2018) Combinatorial double auction-based resource allocation mechanism in cloud computing market. J Syst Softw 137:322– 334. https://doi.org/10.1016/j.jss.2017.11.044
- Mashayekhy L, Nejad M, Grosu D (2015) Physical machine resource management in clouds: a mechanism design approach. IEEE Trans Cloud Comput 3:247–260. https://doi.org/10.1109/TCC.2014.2369419
- 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, p 1–8. https://doi.org/10.1109/ICCCN.2019.8846922
- Dibaj S, Miri A, Mostafavi S (2020) A cloud priority-based dynamic online double auction mechanism (pb-dodam). J Cloud Comput 9:1–26. https:// doi.org/10.1186/s13677-020-00213-7
- Umer A, Nazir B, Ahmad Z (2022) Adaptive market-oriented combinatorial double auction resource allocation model in cloud computing. J Supercomput 78(1):1244–1286. https://doi.org/10.1007/s11227-021-03918-x
- Ullah Z, Umer A, Zaree M, Ahmad J, Alanazi F, Amin NU, Umar Al, Jehangiri Al, Adnan M (2021) Negotiation based combinatorial double auction mechanism in cloud computing. CMC-Comput Mater Contin 69:2123– 2140. https://doi.org/10.32604/cmc.2021.015445
- Lee YC, Zomaya AY (2012) Energy efficient utilization of resources in cloud computing systems. J Supercomput 60(2):268–280. https://doi.org/10. 1007/s11227-010-0421-3
- 24. Yadav R, Zhang W, Kaiwartya O, Singh PR, Elgendy IA, Tian Y-C (2018) Adaptive energy-aware algorithms for minimizing energy consumption and SLA violation in cloud computing. IEEE Access 6:55923–55936. https://doi.org/10.1109/ACCESS.2018.2872750
- Kumar D, Baranwal G, Raza Z, Vidyarthi DP (2018) A truthful combinatorial double auction-based marketplace mechanism for cloud computing. J Syst Softw 140:91–108. https://doi.org/10.1016/j.jss.2018.03.003
- Baranwal G, Kumar D, Raza Z, Vidyarthi DP (2018) Double auctionbased cloud resource provisioning. In: Auction Based Resource Provisioning in Cloud Computing, p 75–95. https://doi.org/10.1007/ 978-981-10-8737-0_5
- Sandholm T, Suri S (2003) Bob: improved winner determination in combinatorial auctions and generalizations. Artif Intell 145(1–2):33–58. https:// doi.org/10.1016/S0004-3702(03)00015-8
- Sandholm T, Suri S, Gilpin A, Levine D (2005) Cabob: a fast optimal algorithm for winner determination in combinatorial auctions. Manage Sci 51(3):374–390. https://doi.org/10.1287/mnsc.1040.0336
- Lehmann D, Oćallaghan LI, Shoham Y (2002) Truth revelation in approximately efficient combinatorial auctions. JACM 49(5):577–602. https://doi. org/10.1145/585265.585266
- Zaman S, Grosu D (2013) Combinatorial auction-based allocation of virtual machine instances in clouds. J Parallel Distrib Comput 73(4):495–508. https://doi.org/10.1016/j.jpdc.2012.12.006
- 31. 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. https://doi.org/10.1109/TPDS. 2014.2308224

- Samimi P, Teimouri Y, Mukhtar M (2016) A combinatorial double auction resource allocation model in cloud computing. Inf Sci 357:201–216. https://doi.org/10.1016/j.ins.2014.02.008
- Vickrey W (1961) Counterspeculation, auctions, and competitive sealed tenders. J Financ 16(1):8–37. https://doi.org/10.2307/2977633
- Li L, Liu YA, Liu KM, Ma XL, Yang M (2009) Pricing in combinatorial double auction-based grid allocation model. J China Univ Posts Telecommun 16(3):59–65. https://doi.org/10.1016/S1005-8885(08)60228-9
- Di S, Wang C-L, Chen L (2013) Ex-post efficient resource allocation for selforganizing cloud. Comput Electr Eng 39(7):2342–2356. https://doi.org/10. 1016/j.compeleceng.2012.12.018
- 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. https://doi.org/10.1016/j.jss.2016.12.009
- 37. McAfee RP (1992) A dominant strategy double auction. J Econ Theory 56(2):434–450. https://doi.org/10.1016/0022-0531(92)90091-U
- Cramton P, Shoham Y, Steinberg R (2006) Combinatorial Auctions. MIT Press, Boston

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