Online Combinatorial Double Auction for Mobile Cloud Computing Markets

Ke Xu*, Yuchao Zhang*, Xuelin Shi*, Haiyang Wang†, Yong Wang‡ and Meng Shen§
*Department of Computer Science and Technology, Tsinghua University, Beijing, China
Email: xuke@mail.tsinghua.edu.cn, zhangyc14@mails.tsinghua.edu.cn, shixuelin@sina.com
†Department of Computer Science, University of Minnesota at Duluth, USA
Email: haiyang@d.umn.edu
‡School of Social Sciences, Tsinghua University, Beijing, China
Email: wang.yong@tsinghua.edu.cn
§School of Computing Science, Institute of technology, Beijing, China
Email: shenmengnetlab@gmail.com

***Abstract*—The emergence of cloud computing as an efficient means of providing computing as a form of utility can already be felt with the burgeoning of cloud service companies. Notable examples including Amazon EC2, Rackspace, Google App and Microsoft Azure have already attracted an increasing number of users over the Internet. However, due to the dynamic behaviors of some users, the traditional cloud pricing models cannot well support such popular applications as Mobile Cloud Computing (MCC). To mitigate this problem, we take our fist steps towards the design of an efficient double-sided combinatorial auction model in the context of mobile cloud computing. In particular, we carefully develop the framework of online combinatorial double auctions and apply a Winner Determination Problem (WDP) model for the proposed auction mechanism. The experiment results indicate that the allocation efficiency of our proposed online auction mechanism is comparable to the social optimal solution.**

I. Introduction

Cloud computing is emerging as a promising paradigm that enables on-demand and elastic access to computing infrastructures. Despite the burgeoning of Internet cloud services, the existing cloud markets are still in the premature stages with respect to their pricing structures. Amazon EC2, for example, advertises $\$0.03-0.12$ per hour for each of its Virtual Machine (VM) instances, depending on their types. Such a posted-offer pricing model is commonly used when the commodity to be priced has a well-known value that is common knowledge to both sellers and buyers, and a buyer is simply a price-taker that chooses whether or not to pay the price, complete the transaction, and acquire the commodity. As a price taker, a buyer cannot affect the price of the commodity. Such a fixed pricing scheme, while perhaps acceptable to a small group of enterprise and individual users, essentially shut the door upon the vast majority of potential cloud users.

To mitigate such a problem, the auction-based instances are widely suggested in the cloud market. Such *Spot Instances* allow the customers to bid on unused resources (e.g., EC2 VMs) and run those instances as long as their bids exceed the current spot price, bringing more freedom to users. Researchers therefore proposed different auction mechanisms to implement resource allocation and pricing in cloud markets [1] [2] [3] [4]. However, these single-sided single-minded auction models cannot well support such popular cloud applications as Mobile Cloud Computing (MCC). In particular, Sharrukh Zaman added detailed reasons in [5] that auctions have clear advantages over others when the auctioned items are complementary. In other words, the auctioned items will have a higher value as a set than as separate parts. It is known that the usage of mobile computing is increasing rapidly. A survey from Juniper Research [6] states that the consumer and enterprise market for cloud-based mobile applications is expected to mount to *\$9.5* billion by 2014. It is thus important to develop a smarter auction model to support such an elevating demand.

In this paper, we for the first time introduce an efficient double-sided combinatorial auction model in the context of mobile cloud computing. To better support the MCC applications and users, we carefully design the framework of online combinatorial double auctions and apply a WDP model for the proposed auction mechanism. We further develop a decomposition algorithm to solve WDP, which can effectively determine winners as well as prices of each auction in affordable time. Moreover, we also investigate a bidding language to facilitate mobile users to express valuations concisely. Our experiment results show that the allocation efficiency of our proposed online auction mechanism is comparable to the social optimal solution and computationally feasible.

The rest of this paper is organized as follows: Section 2 reviews some related work, such as combinatorial and double auctions. Section 3 proposes a framework of the MCC combinatorial double auction. Section 4 describes our novel bidding language $\mathcal{L}_{MU}$ for mobile users. The model and algorithm of WDP for our auction mechanism are presented in Section 5. Then the simulation results are given in Section 6. Finally, Section 7 concludes the paper.

II. Related Work

As cloud computing is designed to be a market-oriented computing paradigm, resource allocation and pricing are always hot topics. Bidding and auctions are deemed to be effective solutions to grid and computing resource markets [7].

As MCC is the combination of wireless access services and cloud services, it is feasible to apply auctions in MCC markets. In this section, we will present related work on auctions (especially combinatorial and double auctions), and then review the use of auctions to implement resource allocation and pricing in cloud and MCC markets.

A. Combinatorial and Double Auctions

Auctions are effective economic ways for setting the price of commodities based on supply and demand in real-world markets. The auction model supports one-to-many (for instance, single-sided auction) or many-to-many (e.g., double auction) negotiations between sellers and buyers, and reduces negotiations to a single value (i.e., price). Moreover, in an auction, players may be allowed to bid for one commodity or sets of items at one time. Therefore, the design of auction mechanism is the hot spot in micro-economics.

Combinatorial Auctions (CAs) that allow bids for bundles of items, provide a great way of allocating multiple distinguishable items among bidders whose perceived valuations for combinations of those items [8]. CAs can make bidders flexibly reveal their preferences on the replaceable or complementary relationships of items, which can decline the bidding risk, increase revenue and thus improve the economic efficiency of auction remarkably. The main topics concerned in CAs are: bidding languages, winner determination and mechanism design.

In recent years, CAs have generated significant interest as automated mechanisms for buying and selling bundles of scarce resources. The developments of the Internet and e-commerce have provided a wonderful platform for CAs. CAs have been applied in wide economic domains successfully, including truckload transportation, industrial procurement, radio spectrum auction, airport time slots, etc.

Now many e-commerce platforms adopting CAs only support one-to-many negotiations, i.e., single-sided auctions. One auctioneer initials an auction before many buyers bid, and vice versa. Although single-sided auctions are well-suited for markets with a limited number of buyers or sellers, these mechanisms are non-effective when the markets consist of numerous of buyers and sellers. To maximize the profits, a potential buyer or seller may bid repeatedly in various auctions, and he/she has to contemplate the possible outcomes of the auctions hosted by different auctioneers. This computational burden hinders the trades, especially in CAs. To relieve this computational burden and promote transactions, many recent researches have been devoted to the double auctions [9].

Double auctions are many-to-many negotiations, which enable multiple buyers and sellers to bid simultaneously in one auction. Indeed, the major exchanges today, like NASDAQ, New York Stock Exchange (NYSE) and the major foreign exchange (FX), apply variants of double auctions [10].

B. Auctions in cloud and MCC Markets

Auction-based mechanisms have been proposed in various fields such as network bandwidth, wireless spectrum, energy industries and advertisements, which investigate how participants behave in a competition for resources. The use of auctions in computing dates back to 1968 when Sutherland [11] proposed allocating processing time in a single computer via auctions. Then with the development of grid computing, many market-based resource allocation strategies were brought out, some of which applied auction mechanisms to grid scheduling [7] [12].

Cloud computing appeared as a more effective market-oriented computing paradigm than grid computing, so currently researchers are investigating the economic aspects of cloud computing from different points of view. Buyya *et al.* [1] proposed an infrastructure of federated clouds for auction-based resource allocation across multiple clouds. Prasad *et al.* [13] and Zaman *et al.* [5] used combinatorial auctions to implement computing resources and virtual machine allocation. Furthermore, Lee *et al.* [2] brought out a real-time group auction system for cloud application allocation. Zhang *et al.* [3] put forward a framework for online auctions in cloud computing. In our former work [4], a continuous double auction mechanism was proposed for cloud markets, and a bidding strategy was designed for cloud users and CSPs to maximize their profits.

For MCC resource and application allocation, there is little work introducing auction mechanisms to MCC markets. Niyato *et al.* [14] developed an auction mechanism with premium and discount factors for resource allocation in MCC systems. The major difference between our work and the current work is that we are considering a combinatorial double auction mechanism for MCC markets, which enables mobile users and MCC providers to submit bids and asks simultaneously and supports users to bid sets of commodities at one time.

III. The Framwork of MCC Combinatorial Double Auction

We consider a platform for MCC markets where multiple mobile users and MCC providers respectively buy and sell commodities in a combinatorial double auction manner. Our solution is efficient for resource allocation in MCC and appeals to mobile users and MCC providers. On one hand, mobile users can bid bundles of applications and services on the platform with light mobile data usage. On the other hand, providers can supply sets of commodities in each auction.

A. Design Requirements

A feasible auction model for MCC should meet the following requirements:

Firstly, the MCC services are the combinations of wireless services and cloud computing resources. As shown in Fig.1, the mobile users access services provided by remote clouds via wireless networks, like 2G, 3G or WiFi. Mobile communication base stations or WiFi access points provide radio resources (i.e., bandwidth), while remote cloud provide applications, computing and storage resources. Obviously, if mobile users want to use cloud services, they also need to buy wireless access services. In a feasible MCC market, commodities cover

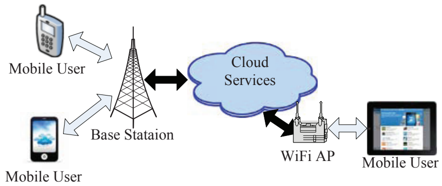


Fig. 1. MCC enabling mobile users accessing cloud services through wireless networking

wireless access services as well as applications, computing and storage resources. Therefore, an effective auction mechanism should allow each user to submit a bid for sets of items (sometimes called bundles), rather than to bid every item in many sequential auctions.

Secondly, energy efficiency is of particular importance for mobile devices. Moreover, both transmission and computation consume energy. In wired clouds, this is not a big concern. To attract mobile users to take part in auctions, a mechanism allows users to submit bids while transmitting data as few as possible with few computations to construct their bids. Furthermore, mobile users access online auction platforms roaming across different wireless networks, so the less data transmission gets the lower cost. Therefore, a concise bidding language is vital for mobile users.

Thirdly, different from traditional CSPs, MCC providers usually offer various applications besides computing utilities and storage resources, such as image processing, natural language translating, and multimedia search [15]. As mentioned previously, mobile users are attracted to buy a package of commodities together. For example, an increasing number of users prefer to take photos using mobile devices. However, because of the limited storage space they prefer to upload some photos to online storage servers when there are inadequaete space in their mobile devices. An MCC provider *P1* supplies such storage services. To get additional profits, *P1* also provides an animator application and other image-processing applications. Thus, a mobile user *Peter* may buy 1GB storage space and 10 runs animator for 1 year. Thus when *Peter* takes a new photo, he can upload it to the storage servers of *P1*. If he wants to make an animation, he can select photos on the servers and submit them to the animator application. The application will run on the remote servers and return the result to *Peter*. Because there are more and more such applications in MCC markets, the scale of one auction may be large. Consequently, an effective combinatorial auction mechanism is vital for MCC markets, which should quickly determine winners and prices of an auction consisting of many users and providers.

Finally, in current cloud markets cloud users often rent resources to support websites, or run scientific computing [16] [17]. They are professional enough to accept the complicated online auctions, and they even can leave bidding to the user brokers. On the other hand, the simple auction rules are more acceptable to mobile users.

In summary, in such competitive MCC markets populated by mobile users and MCC providers, combinatorial auctions (CAs) are feasible to solve resource allocation and pricing problems. Each mobile user demands sets of various commodities, who can bid bundles in one combinatorial auction. Moreover, double auctions can also be adopted to improve market efficiency, in which buyers and sellers can submit bids simultaneously.

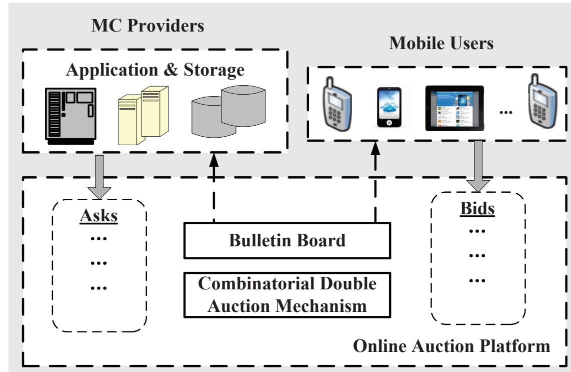


Fig. 2. A Framework of the MCC Combinatorial Double Auction Platform

The first difference between our solution and the current ones is that we consider a combinatorial double auction mechanism for MCC markets, which enables mobile users and MCC providers to submit bids and asks simultaneously, and supports users to bid sets of commodities at one time. The second difference is that we design a novel bidding language for mobile users.

B. Framework Overview

The online auction platform collects the bids and asks from mobile users and MCC providers respectively, and it computes who and how to win the auctions. The overview of the framework is shown in Fig.2.

It is an electronic bidding platform, which can be easily accessed via the Internet and make use of e-commerce technologies. The platform plays the role of an auctioneer, on which mobile users can submit bids for bundles of items, while MCC providers can submit asks. In addition, auctions are all carried out in an online manner, i.e., users and providers can take part in auctions whenever they need, and the platform can determine winners and price instantaneously as soon as auctions close.

The auction on the platform can be divided into 3 states: the registration stage, the bidding stage and the winner determination stage. In the registration stage, the information of all resources, the related parameters of mobile users, and MCC providers are presented on the bulletin board, and every player is certified. Then in the bidding stage, users can submit bids and providers can submit asks. Eventually, the winners and prices are computed by the determination module according to the combinatorial double auction mechanism.

Furthermore, mobile users can download a bidding application, which executes on mobile devices to translate user's specific demands into requests described in the bidding language, by which user's heterogeneous demands can be restricted to regulated and consistent forms while the details of the requirements can still be revealed. Each request is then submitted to the platform. MCC providers also can submit asks of commodities they want to sell. After an auction closes, the platform computes winners and prices based on the auction mechanism, and then announces the results to users and providers who establish the connection and start to run/host applications once the charging and payment are complete.

The bidding rules of the platform are given in the next subsection. Section 4 describes the bidding language for mobile users, and Section 5 presents how to determine winners in the MCC combinatorial double auction.

C. The Market Rules

In our MCC auction framework, the platform acts as a central auctioneer who receives the bids and asks, and then carries out all the computation to find the optimal allocation of items to bidders. To facilitate bidders and improve trading efficiency, some market-rules are defined as follows.

Rule 1: The platform prescribes the **Bidding Period**, $\boldsymbol{t_{bp}}$, can be one day, several hours, etc. During the Bidding Period, mobile users and MCC providers are allowed to submit bids and asks, by the end of which the auction closes and the market clears. At an auction, only one bid or ask can be submitted by each mobile user or MCC provider. At the end of the Bidding Period, all bids and asks are opened. Furthermore, auction results i.e, winners and prices are also published.

Rule 2: A bid of user i can be for bundles of items, denoted as $\boldsymbol{B_i = \mathcal{L}_{MU}(< S, v_i^S >)}$. $\boldsymbol{S}$ is a subset of the available commodities in the auction. v_i^S is a valuation (willingness to pay) of user i for $\boldsymbol{S}$. Each mobile user usually has heterogenous demands and valuations for commodities, which can be expressed by the bidding language $\mathcal{L}_{MU}$.

Rule 3: An ask of provider j can be for multiple units of items, denoted as $\boldsymbol{A_j = (< r_1, c_j^{r_1}, q_j^{r_1} >, \ldots, < r_k, c_j^{r_k}, c_j^{r_k} >, \ldots, < r_m, c_j^{r_m}, q_j^{r_m} >)}$. $c_j^{r_k}$ is the offered price per unit for commodity r_k of provider j, and $q_j^{r_k}$ is the quantity of commodity r_k.

Rule 4: To prevent unreasonably low bids and speed up the trading process, the **Minimum Bid** allowed in the market, $\boldsymbol{B_{min}}$, is defined. It can be set according to history transaction prices or 0. For any atomic component of each bid B_i, $v_i^S/|S| \geq B_{min}$.

Rule 5: In the same way, to prevent unreasonably high asks and speed up the trading process, the **Maximum Ask** allowed in the market, $\boldsymbol{A_{max}}$, is defined. It can also be set according to history transaction records or $+\infty$. For any element in each ask A_j, $c_j^{r_k} \leq A_{max}$.

The above rules are published on the MCC auction platform. As long as users and providers take part in auctions, they must submit bids and asks according to the rules. The rules not only ensure auction efficiency, but also enable users and providers to understand the auction mechanism. In addition, the history transaction records published on the the platform help users to decide their valuations on various services and applications.

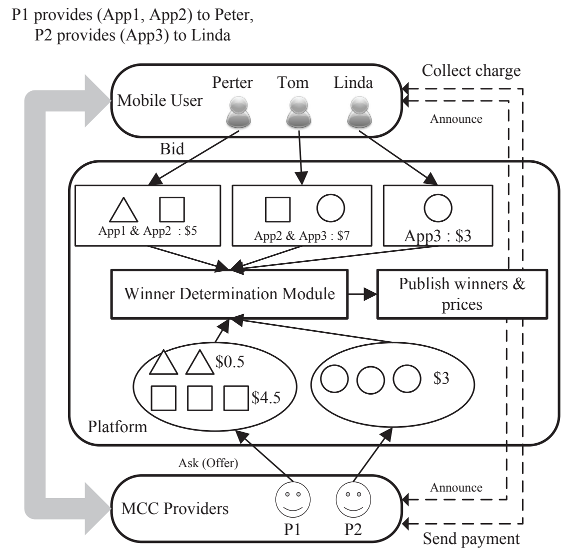


Fig. 3. A scenario of an auction: consisting of 2 providers and 3 users bidding for 3 commodities

D. Scenario of A Combinatorial Double Auction on The Platform

With the online auction platform and bidding rules, mobile users and MCC providers can trade by auctions. Users often have a variety of demands, and providers also supply various services and applications. For a mobile user who needs two applications, bidding two items in one combinatorial auction is more efficient than in two sequential auctions separately. Moreover, double auctions allows users and providers to bid simultaneously in one auction, which also improves market efficiency. Advantages of the combinatorial double auction for MCC markets can be revealed in the following scenario shown in Fig. 3.

During the bidding period, mobile users and MCC providers may submit bids and asks simultaneously. Each user can bid one or bundles of items, while each provider can supplies multiple units of commodities. In Fig. 3, *Peter* seeks to buy one unit of *App1* and *App2* before bidding *$5*. *Tom* bids *$7* for one unit of *App2* and *App3*. *Linda* bids *$3* for one unit of *App3*. All the three users submit their bids to the online auction platform. Two MCC providers, *P1* and *P2*, take part in the auction. *P1* sells two units of *App1* at price *$0.5* and three units of *App2* at price *$4.5*. *P2* sells three units of *App3* at price *$3*. Both providers also submit their asks to the platform.

At the end of the bidding period, the winner determination module computes winner and prices of the auction, the algorithm of which will be described in Section 5. Then the results are announced to users and buyers. In the auction, *Peter* wins one unit of *app1* and *app2* supplied by *P1*, and *Linda* wins one unit of *$3* supplied by *P2*. However, *Tom* loses to *Peter* and *Linda*.

As shown in the above scenario, the combinatorial double auction mechanism is effective and flexible for mobile users and MCC providers. It not only ensures competitive bidders and offers simultaneously, but also allows users to bid for bundles of items at one time. However, to apply the auction mechanism to real markets, there are two key problems to solve: a concise bidding language for users and a feasible WDP algorithm. The solutions will be given in the following sections.

IV. The Bidding Language for Mobile Users

Bidding language is a language for expressing valuation functions, and a good one which allows bidders to concisely express natural valuation functions. In our MCC auction framework, the bidding language is implemented in the mobile client side to translate user's specific demands into requests. In this section, we first analyze different types of mobile user valuations, and then we put forward a novel bidding language $\mathcal{L}_{MU}$ to represent heterogenous user demands concisely and consistently. At last, we discuss the novel contributions of $\mathcal{L}_{MU}$ in MCC combinatorial double auctions.

A. Heterogenous Mobile User Valuations

In general, let $\boldsymbol{R}$ be the set of all the types of goods for sale in a CA, a buyer could have a different valuation for every subset $\boldsymbol{S}$ of $\boldsymbol{R}$. Because $\boldsymbol{R}$ has $2^{|R|}-1$ different subsets, there are $2^{|R|}-1$ possible bids to specify in the CA.

Furthermore, how valuable one item is to a buyer may depend on whether he/she possesses another item. On one hand, some of these items are substitutable (e.g., users can use storage from different places) that they have similar functions to the users. On the other hand, some items are complementary that users will need them as a bundle (e.g., users need both wireless connection and storage for online photo posting).

The complementary and substitutable items in an MCC market can be defined as follows.

Definition 1: A mobile user i has a valuation for a commodity r, denoted as $v_i^{\{r\}}$. For user i, items a and b are **substitutable** if $v_i^{\{a,b\}} < v_i^{\{a\}} + v_i^{\{b\}}$, and items a and b are **complementary** if $v_i^{\{a,b\}} \geq v_i^{\{a\}} + v_i^{\{b\}}$. Especially, when items a and b are independent, are also be viewed as complementary because $v_i^{\{a,b\}} = v_i^{\{a\}} + v_i^{\{b\}}$.

The different valuations for various items lie on user's utilities. A user is satisfactory with the allocated resources, which is referred to as the utility. Because of the complementarity and substitutability, a mobile user's total utilities do not always equal to the sum of his/her utility of each commodity. An efficient auction mechanism should maximize buyer's utilities and seller's payoffs, so does our MCC mechanism. Thus we formulate the user utility functions as follow:

$$U_i(S) = v_i^S - \sum_{r \in S} P_i^r \tag{1}$$

where $U_i(S)$ is the utility of user i on the commodity set S, v_i^S is the valuation of S, and P_i^r is the transaction price on which user i gets the item r. The utility function can reveal the complementarity and substitutability because user's valuations can express them, i.e.,

$$U_i(\{a,b\}) = \begin{cases} U_i(\{a\}) + U_i(\{b\}) + h_i \\ \qquad \textit{for a, b is complementary} \\ U_i(\{a\}) + U_i(\{b\}) - l_i \\ \qquad \textit{for a, b is substitutable} \end{cases} \tag{2}$$

In (2), $h_i \geq 0$ and $l_i \geq 0$ can be viewed as the premium and discount in one auction respectively. From the standpoint of the buyer, if he/she can buy two complementary items a and b in one auction successfully, he/she is willing to pay more. However, two substitutable items are bought at one time only when he/she can get a discount.

To express such heterogenous user demands in CAs, many bidding languages were brought up, which are meant to provide the syntax for encoding bid's information in a succinct and simple manner. Similar to any language, there is a trade-off between expressiveness and simplicity. In the next subsection, we review these bidding languages and put forward our novel language to express heterogeneous demands of mobile users in MCC markets.

B. Semantics of Bidding Languages

Bidding languages basically try to efficiently model different bidding patterns. The most common method is the single-minded bidding language, or called atomic bidding language. It can only describe user demands as follow: a user i chooses $\boldsymbol{S}$, a subset of available items $\boldsymbol{R}$, for valuation v_i^S [18].

Obviously, the single-minded bids are not expressive enough to distinguish complementarity and substitutability, so the OR, XOR, and other bidding language are put forward. In OR, bundle-value pairs are ORed together, and any number of these pairs may be accepted in an auction. For example, $(\{a\},3)OR(\{b,c\},4)$ implies a value of 3 for $\{a\}$ and a value of 7 for $\{a,b,c\}$. OR is good for expressing complementarity, but bad for expressing substitutability. XOR can express any valuation function, which simply XOR together all bundle-value pairs. It means that only one of the bundle-value pairs can be accepted in an auction. For example, $(\{a\},3)XOR(\{b,c\},4)$ implies a value of 3 for $\{a\}$ and a value of 4 for $\{a,b,c\}$. While XOR is more expressive than OR, there are valuations that can be specified more succinctly by OR. However, they sometimes are not very concise, therefore some solutions try to combine OR and XOR to get benefits of the both. The result introduces new languages: OR-of-XORs, XOR-of-ORs, and the logical language [19].

Often each language is good in expressing some patterns and weak or unable in expressing some other patterns. While certain languages can be compared based on their expressivity, it is not always possible to accurately compare two bidding languages. However, the complicated bidding languages are obviously more efficient to express various combinatorial bids than the simple languages (atomics, OR and XOR), while the former are more expensive on computing costs of WDP than the latter.

If our online MCC auction platform adopts a complicated bidding language, although it allows mobile users to submit many kinds of combinatorial bids, the performance of the platform will still be good. Because in MCC markets, mobile users are non-professional traders, they often cannot design multiple combinations of various bids. Furthermore, if the auction platforms are efficient enough to implement many transactions immediately at the end of bidding period with acceptable costs, the auctions can be held frequently and the users do not need to bid many goods in one auction. Therefore, our novel bidding language $\mathcal{L}_{MU}$ restricts the kinds of combinations that bidders may bid on, which not only transmits this bidding function in a succinct way to the platform but also reduces computational complexity.

The semantics of an $\mathcal{L}_{MU}$ bid can be expressed in Backus-Naur Form (BNF) as follows:

$$BID ::= (Comb\_Bid)|(Comb\_Bid)^{\leq n}$$
$$Comb\_Bid ::= Atom\_Bid|Atom\_Bid{\rightarrow}Atom\_Bid$$
$$Atom\_Bid ::= < S, v^S >$$

Therefore, an $\mathcal{L}_{MU}$ bid can be one of four forms:

1) **An atomic bid**, $(< S, v^S >)$, means a user bids a set of commodities S ($S \subseteq R$) with the valuation v^S ($v^S \in \mathcal{N}$, and $v^S/|S| \geq b_{min}$). It is same as a single-minded language, which can express complementarity.
2) **A combinatorial bid**, i.e., two atomic bids joined by a binary operator $\rightarrow$, is denoted as $(< S_1, v^{S_1} >\rightarrow< S_2, v^{S_2} >)$, where $S_1 \subset R$, $S_2 \subset R$, and $S_1 \bigcap S_2 = \phi$. When a mobile user wants to bid substitutable goods, he/she can express in the form. The equivalent representation of this form in our $\mathcal{L}_{MU}$ and XOR language are:
$$(< S_1, v^{S_1} >\rightarrow< S_2, v^{S_2} >) \Longleftrightarrow (< S_1, v^{S_1} > XOR < S_2 \cup S_1, v^{S_2} >) \quad (3)$$
The user may be allocated S_1 or $S_2 \cup S_1$, but the two cannot appear simultaneously.
3) **An atomic bid with quantity range**, $(< S, v^S >)^{\leq n}$, means a user wants to buy the atomic bid up to n units ($n \in N$, and $n > 1$).
4) **A combinatorial bid with quantity range**, $(< S_1, v^{S_1} >\rightarrow< S_2, v^{S_2} >)^{\leq n}$, means a user wants to buy the combinatorial bid up to n units ($n \in N$, and $n > 1$).

The first two forms are suitable for mobile users who just buy one unit of each type commodity, and the latter two allow users to buy n copies of the same bid. Let *Bid* is one atomic or combinatorial bid in $\mathcal{L}_{MU}$, multi-units of *Bid* can be represented by **OR** as follows:

$$Bid^{\leq n} \Longleftrightarrow \underbrace{(Bid\,\boldsymbol{OR}\,Bid\,\boldsymbol{OR}\ldots\boldsymbol{OR}\,Bid)}_{n} \quad (4)$$

The user can get one group of goods described in *Bid*, or *2* groups, at most n groups.

In Fig.3, the demand users *Peter*, *Tom* and *Linda* all submit atomic bids: $B_P = (< \{1,2\}, \$5 >)$, $B_T = (< \{2,3\}, \$7 >)$ and $B_L = (< \{3\}, \$3 >)$. They also can combine atomic bids to combinatorial bids or multi-units according to their demands.

C. Advantages over Previous Languages

The design for bidding languages involves a trade-off between expressiveness and simplicity, so our $\mathcal{L}_{MU}$ also needs to express heterogenous demands of mobile users in a concise way. Compared with OR, XOR and other complicated logical bidding languages, $\mathcal{L}_{MU}$ has the following advantages:

(1) **Ease of use**: The semantics of $\mathcal{L}_{MU}$ is easy to understand, and bidders can express their demands in the correct format of $\mathcal{L}_{MU}$. The general mobile users cannot handle too many logical operators, such as OR, XOR, and AND. Therefore, they prefer to submit simple bids rather than apply various logical operators to combine bids.

Consider the following scenario. There are *3* types of services to auction, storage services, GIF animator services and Flash maker services, denoted as a, b, and c respectively. A user, *Peter* submits an atomic bid: $B = (< \{a, b\}, \$5 >)$, which means he wants to buy one unit of storage service and GIF animator service, which is valued complementary. If *Peter* expects he has many pictures to be saved and processed to GIF, he can submit $(< \{a, b\}, \$5 >)^{\leq 3}$ which means he can get *3* copies of them at most. Furthermore, *Peter* deems GIF animator service and Flash-maker service are substitutable, and he buys both services only when there will be a discount. He can submit a combinatorial bid $(< \{a, b\}, \$5 >\rightarrow< \{c\}, \$6 >)$.

(2) **Representing quantity ranges**: The previous bidding languages cannot express buyer's demands for multi-units of goods directly. If a buyer needs *3* units of the good a at most, he can submit a bid denoted as $< \{a\}, \$2 > OR < \{a\}, \$2 > OR < \{a\}, \$2 >$ in OR. His demand even can not be expressed in XOR. However, our $\mathcal{L}_{MU}$ provides a simple way to represent quantity ranges.

(3) **Conciseness**: $\mathcal{L}_{MU}$ is concise in two respects. Firstly, the quantity ranges of user's demands can be represented simply. Secondly, applying a binary operator $\rightarrow$ to express a bid consisting of substitutable goods needs less characters than that used in XOR. For example, *Peter* deems GIF animator service and Flash-maker service are substitutable, so his bid expressed in $\mathcal{L}_{MU}$ is $(< \{a, b\}, \$5 >\rightarrow< \{c\}, \$6 >)$, while in XOR is $(< \{a, b\}, \$5 > XOR < \{a, b, c\}, \$6 >)$

(4) **Low cost of wireless network transmission**: The bids of mobile users are submitted to the auction platform via various wireless networks. The features of $\mathcal{L}_{MU}$ make the bids be described simply, especially expressing quantity ranges and demands for substitutable goods. Therefore, the costs of wireless network transmission are reduced.

V. The Winner Determination Problem

The problem of identifying which set of bids to accept has usually been dubbed the WDP, or the combinatorial allocation problem (CAP), which is a computational problem of how to efficiently determine the item allocation once the bids and asks have been submitted to the auction platform. The efficiency of an auction mechanism depends on the WDP model and its algorithm.

A general WDP model of a single-sided combinatorial auction can be stated as follows: in an auction, given the set $\boldsymbol{R}$ of commodities, the set $\boldsymbol{I}$ of bidders, and the set $\boldsymbol{B}$ of bids submitted by all the bidders, find an item allocation to bidders, which maximizes the auctioneer's revenue. More formally, the model can be denoted as:

$$\begin{aligned} &\max \sum_{i \in I} \sum_{S \subseteq R} B_i(S) x(S,i) \\ &s.t. \sum_{r \in S} \sum_{i \in I} x(S,i) \leq 1 \quad \forall r \in R \\ &\qquad \sum_{S \subseteq R} x(S,i) \leq 1 \quad \forall i \in I \\ &\qquad x(S,i) \in 0,1 \quad \forall S \subseteq R, i \in I \end{aligned} \tag{5}$$

$\boldsymbol{S}$ is a subset of $\boldsymbol{R}$, i.e., $\boldsymbol{S} \subseteq \boldsymbol{R}$. $B_i(S)$ is a bid for $\boldsymbol{S}$ submitted by bidder i. Without loss of generality, let $B_i(S) \geq 0$. If $\boldsymbol{S}$ is allocated to bidder i, $x(S,i) = 1$, otherwise $x(S,i) = 0$.

It is an integer programming problem, which has been proved to be **NP-hard** [20]. This problem is difficult for large set of commodities $\boldsymbol{R}$, specifically if bids exists for all subsets of commodities.

Our solution is a many-to-many auction mechanism, i.e., combinatorial double auction, which allow buyers and sellers bid simultaneously in one auction. The general combinatorial auctions are single-sided, therefore the WDP model described in problem (1) is unsuitable for our MCC combinatorial double auction. Obviously, the objective of such double auctions should be maximizing the total surpluses of all traders, including buyers and sellers. Therefore, the WDP of our auction mechanism is formulated as an optimization problem to maximize the **social welfare**, i.e., the total payoffs/utilities of the users and providers.

In our combinatorial double auction, there are set $\boldsymbol{R}$ of commodities, set $\boldsymbol{I}$ of mobile users, and set $\boldsymbol{J}$ of MCC providers. Given set $\boldsymbol{B} = \{B_1, \ldots, B_i, \ldots, B_{|I|}\}$ of bids submitted by all users, and set $\boldsymbol{A} = \{A_1, \ldots, A_j, \ldots, A_{|J|}\}$ of asks offered by all providers, find an allocation of goods to users, which maximizes the social welfare. To formulate a feasible WDP model for our MCC combinatorial double auction, the bids and asks need to be preprocessed.

A. Preprocessing of Bids and Asks

$\mathcal{L}_{MU}$ enables every user to submit one of four types bids (an atomic bid, an atomic bid with quantity range, a combinatorial bids and a combinatorial bid with quantity range). To simplify WDP, the **dummy goods** and **sub-users** are introduced to transform various bids into one format: the one-unit atomic bid. The bid transformation can be conducted according to the following theorems:

Theorem 1: Any atomic bid with quantity range submitted by a user, $B_i = (< S, v^S >)^{\leq n}$, can be transformed to n atomic bids $(< S, v^S >)$. Suppose they are submitted by n sub-users. The solution to the origin WDP can be obtained by the solution to the new WDP.

Theorem 2: Any combinatorial bid, $B_i = (< S_1, v^{S_1} > \rightarrow < S_2, v^{S_2} >)$, can be transformed to 2 atomic bids by introducing a dummy good ($dummy_i$) and 2 sub-users (su_i^1, su_i^2). Suppose su_i^1 submits $(< S_1 \cup dummy_i, v^{S_1} >)$ and su_i^2 submits $(< S_1 \cup S_2 \cup dummy_i, v^{S_2} >)$. The solution to the origin WDP can be obtained by the solution to the new WDP.

Theorem 3: Any combinatorial bid with quantity range, $B_i = (< S_1, v^{S_1} > \rightarrow < S_2, v^{S_2} >)^{\leq n}$, can be transformed to $2 \times n$ atomic bids. The solution to the origin WDP can be obtained by the solution to the new WDP.

In the same way, **sub-providers** are also introduced to simplify the asks offered by the MCC providers. The transformation of asks can be conducted according to the following theorem:

Theorem 4: Any ask offering more than one type of goods, $(< r_1, c_j^{r_1}, q_j^{r_1} >, < r_2, c_j^{r_2}, q_j^{r_2} >, \ldots, < r_m, c_j^{r_m}, q_j^{r_m} >)$, can be transformed to m simple asks $(< r_m, c_j^{r_m}, q_j^{r_m} >)$. Suppose they are submitted by m sub-providers. The solution to the origin WDP can be obtained by the solution to the new WDP.

The preprocessing is shown in Algorithm 1.

After the origin bids and asks are transformed according to the above theorems, each user/sub-user only submits an atomic bid and each provider/sub-provider only offers an ask of one type of goods. The new commodity set is denoted as $\hat{\boldsymbol{R}}$, which also includes the dummy goods. Similarly, there are new sets $\hat{\boldsymbol{I}}$ of buyers, $\hat{\boldsymbol{J}}$ of sellers, $\hat{\boldsymbol{B}}$ of transformed bids, and $\hat{\boldsymbol{A}}$ of simplified asks. Each item in $\hat{\boldsymbol{B}}$ is denoted as $\hat{B}_i =< S_i, v_i >$, and S_i presents the bundle that the buyer i bids. Similarly, each item in $\hat{\boldsymbol{A}}$ is $\hat{A}_j =< r_j, c_j, q_j >$, and r_j presents the good that the seller j sells.

B. The WDP Model

As the original bids and asks are processed in our MCC combinatorial double auction, the WDP model can be formulated as follows:

$$\begin{aligned} &\max (\sum_{i \in \hat{I}} x_i U_i(S_i) + \sum_{j \in \hat{J}} y_j W_j(r_j)) \\ &s.t. \sum_{i \in \hat{I}, r \in \hat{B}_i(1)} x_i = \sum_{j \in \hat{J}, r = \hat{A}_j(1)} y_j \quad \forall r \in \hat{R} \\ &\qquad y_j \in \{0, 1, \ldots, q_j\} \quad \forall j \in \hat{J} \\ &\qquad x_i \in \{0, 1\} \quad \forall i \in \hat{I} \end{aligned} \tag{6}$$

where $\boldsymbol{x_i}$ denotes whether the buyer i trades in the allocation, and $\boldsymbol{y_j}$ denotes the transaction quantity of the seller j. The variables $(x_i, y_j), i \in \hat{I}, j \in \hat{J}$ specify the auction result. By mapping sub-users and sub-providers to origin mobile users and MCC providers respectively, the MCC resource allocation is acquired.

Algorithm 1 Preprocessing of bids and asks

Input:

1) The set $\boldsymbol{R}$ of the commodities; the set $\boldsymbol{I}$ of the mobile users; the set $\boldsymbol{J}$ of the MCC providers;

2) The set $\boldsymbol{B} = \{B_1, \ldots, B_i, \ldots, B_{|I|}\}$ of bids, the set $\boldsymbol{A} = \{A_1, \ldots, A_j, \ldots, A_{|J|}\}$ of asks

Output:

The set $\hat{\boldsymbol{B}}$ of atomic bids, the set $\hat{\boldsymbol{A}}$ of simple asks, the set $\hat{\boldsymbol{R}}$ of goods and dummy goods, the set $\hat{\boldsymbol{I}}$ of users and sub-users, the set $\hat{\boldsymbol{J}}$ of providers, dummy provider and sub-providers,.

Initialization of $\hat{\boldsymbol{I}}$,$\hat{\boldsymbol{J}}$,$\hat{\boldsymbol{B}}$,$\hat{\boldsymbol{A}} = \varnothing$, $\hat{\boldsymbol{R}} = \boldsymbol{R}$
for all $B_i \in \boldsymbol{B}$ **do**
  if B_i is not an atomic bid **then**
    Transform B_i to a groups of atomic bids S_b
    $\hat{\boldsymbol{B}} = \hat{\boldsymbol{B}} \cup S_b$, $\hat{\boldsymbol{I}} = \hat{\boldsymbol{I}} \cup \{subusers\}$
    for all $dummy_i$ **do**
      $\hat{\boldsymbol{J}} = \hat{\boldsymbol{J}} \cup \{dp_i\}$, $\hat{\boldsymbol{A}} = \hat{\boldsymbol{A}} \cup \{(< dummy_i, 0, 1 >)\}$
      $\hat{\boldsymbol{R}} = \hat{\boldsymbol{R}} \cup \{dummy_i\}$
    end for
  else
    $\hat{\boldsymbol{B}} = \hat{\boldsymbol{B}} \cup B_i$, $\hat{\boldsymbol{I}} = \hat{\boldsymbol{I}} \cup \{user_i\}$
  end if
end for
for all $A_j \in \boldsymbol{A}$ **do**
  if $|A_j| \geq 1$ **then**
    Transform A_j to a groups of simple asks S_a
    $\hat{\boldsymbol{A}} = \hat{\boldsymbol{A}} \cup S_a$, $\hat{\boldsymbol{J}} = \hat{\boldsymbol{J}} \cup \{subproviders\}$
  else
    $\hat{\boldsymbol{A}} = \hat{\boldsymbol{A}} \cup A_j$, $\hat{\boldsymbol{J}} = \hat{\boldsymbol{J}} \cup \{provider_j\}$
  end if
end for

The object of (6) is to maximize the total utilities of the mobile users and MCC providers, i.e., social welfare, denoted as $Z(\boldsymbol{x}, \boldsymbol{y})$. An auction mechanism is efficient if the allocation maximizes social welfare. $U_i(S)$ is the utility function of the buyer i, which has been defined in (1). $W_j(r)$ is the surplus function of the seller j, and is formulated as follow:

$$W_j(r) = P_j^r - c_j^r \tag{7}$$

where P_j^r is the transaction price on which the seller j sells the item r, and c_j^r is the offered price submitted by the seller j. The seller j can obtain the surplus $W_j(r)$ by selling one unit of commodity r.

Therefore, the object of (6) can be presented as:

$$Z(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i \in \hat{I}} x_i (v_i - \sum_{r \in S_i} P_i^r) + \sum_{j \in \hat{J}} y_j (P_j - c_j) \tag{8}$$

Because

$$\sum_{i \in \hat{I}} x_i \sum_{r \in S_i} P_i^r = \sum_{j \in \hat{J}} y_j P_j \tag{9}$$

we have

$$Z(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i \in \hat{I}} x_i v_i - \sum_{j \in \hat{J}} y_j c_j \tag{10}$$

Therefore, our WDP problem can be solved by the following integer program:

$$\begin{aligned} (IP)\ z_{IP} = & \max(\sum_{i \in \hat{I}} v_i x_i - \sum_{j \in \hat{J}} c_j y_j) \\ & s.t. \sum_{i \in \hat{I}} b_{ri} x_i - \sum_{j \in \hat{J}} a_{rj} y_j = 0 \quad \forall r \in \hat{R} \\ & y_j \in \{0, 1, \ldots, q_j\} \quad \forall j \in \hat{J} \\ & x_i \in \{0, 1\} \quad \forall i \in \hat{I} \end{aligned} \tag{11}$$

To present the first constraint clearly, two matrixes $\boldsymbol{b}$ and $\boldsymbol{a}$ are used. The $\boldsymbol{b}$ is a $|\hat{R}| \times |\hat{I}|$ matrix, and each element is *0* or *1*, i.e. $b_{ri} \in \{0, 1\}$. Because all the origin bids are transformed to the atomic bids, one commodity appears at most once in each atomic bid. If buyer i bids the commodity r, $b_{ri} = 1$. Otherwise, $b_{ri} = 0$. Similarly, $\boldsymbol{a}$ is a $|\hat{R}| \times |\hat{J}|$ matrix, and each element is also 0 or 1. If seller j offers the commodity r, $a_{rj} = 1$. Otherwise, $a_{rj} = 0$. Furthermore, there is only one element being *1* in each column of $\boldsymbol{a}$, because all origin asks are transformed to the form $\hat{A}_j =< r_j, c_j, q_j >$ which only consists of one type of commodities. After the origin bids and asks are preprocessed according to the transformation methods, the matrixes $\boldsymbol{b}$ and $\boldsymbol{a}$ are initialized.

In the next subsection, we design the decomposition algorithm to relax the problem $\mathcal{P}$ to a linear formulation, and bring up a pricing mechanism to decide transaction prices.

C. The Decomposition Algorithm And Pricing Mechanism

The optimization problem IP is also NP-hard because it is a special case of the general WDP problem defined in (5), which has been proved to be NP-hard [20]. Therefore, how to find an optimal allocation solution and transaction prices of each commodity is important.

Two approaches are used to find the optimal solution to the general WDP of the single-sided combinatorial auctions, shown in (5). The first one is the exact method, which replaces the given problem by one with a larger feasible region that is more easily solved. The upper bound on the optimal solution value is obtained by solving a relaxation of the optimization problem [20]. The second approach is to conduct one of the standard Artificial Intelligence (AI) searches over all the possible allocations, given the bids submitted [21]. Several algorithms with satisfactory performance for problem sizes and structures occurred in practice have been developed. However, because of the wide applicability of combinatorial auctions, one cannot hope for a general-purpose algorithm that can efficiently solve every instance of this problem. Furthermore, there is little research work on the double combinatorial auctions.

To design a computationally efficient algorithm for our combinatorial double auction problem, we first decompose IP. Our decomposition algorithm reformulates the problem to a linear programming problem, which can then be solved in polynomial time with a subgradient algorithm. Then we rely on the solution to the linear dual problem and use its optimal value to get an optimal solution to the original primal integer program.

We adopt the Lagrangean relaxation to relax the first constraint of our original problem IP by moving it into the objective function with a penalty term. Then we get the Lagrangean relaxation problem LR:

$$\begin{aligned}(LR)\ \ z_{LR}(\boldsymbol{\lambda}) =& \max L(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\lambda}) \\ & s.t.\ 0 \le y_j \le q_j \quad \forall j \in \hat{J} \\ & 0 \le x_i \le 1 \quad \forall i \in \hat{I} \end{aligned} \tag{12}$$

where $L(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\lambda})$ is the Lagrangean function, which is defined as:

$$\begin{aligned} L(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\lambda}) =& \sum_{i \in \hat{I}} v_i x_i - \sum_{j \in \hat{J}} c_j y_j \\ &+ \sum_{r \in \hat{R}} \lambda_r (\sum_{j \in \hat{J}} a_{rj} y_j - \sum_{i \in \hat{I}} b_{ri} x_i) \end{aligned} \tag{13}$$

and $\boldsymbol{\lambda}$ is a vector of Lagrangean multipliers, $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_r, \ldots, \lambda_{|\hat{R}|})$.

Therefore, we get the Lagrangen dual problem LD of the primal problem:

$$\begin{aligned}(LD)\ \ z_{LD} =& \min z_{LR}(\boldsymbol{\lambda}) \\ & s.t.\ \lambda_r \ge 0 \quad \forall r \in \hat{R} \end{aligned} \tag{14}$$

Computing z_{LD} is easy, since there are many subgradient algorithms for the Lagrangean relaxation. Our problem can be deemed as a case of the Traveling Salesman Problem (TSP), and then the subgradient algorithm in [22] is applied. A subgradient of the Lagrangean function $L(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\lambda})$ is defined as:

$$g = \partial L(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\lambda}) / \partial \boldsymbol{\lambda} \tag{15}$$

Iterate $\boldsymbol{\lambda}^{(k)}$ is generated according to the update recursion:

$$\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + t^{(k)} g^{(k)} \tag{16}$$

where $t^{(k)}$ being a scalar representing the step size and $g^{(k)}$ a subgradient of the function $L(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\lambda})$ at the point $\boldsymbol{\lambda}^{(k)}$. Phase I in Algorithm 2 shows the subgradient algorithm in details.

The key point is to decide transaction prices. The constraint $\sum_{i \in \hat{I}} b_{ri} x_i - \sum_{j \in \hat{J}} a_{rj} y_j = 0 (\forall r \in \hat{R})$ restricts that the total demand of all mobile users is equal to the total supply of all MCC providers. Because the vector of the Lagrangean multipliers relaxes it, the $\boldsymbol{\lambda}$ can be interpreted as a price vector. When the Lagrangen dual problem LD is solved, the optimal vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_r, \ldots, \lambda_{|\hat{R}|})$ is also obtained. Therefore, the transaction prices of the commodities are decided, and λ_r is the transaction price of the commodity r. In our MCC combinatorial double auction, one type of goods only have one price. The trade prices of the dummy goods all equal to 0.

Following the above steps, we present our detailed MCC combinatorial auction mechanism in Algorithm 2. It is shown how the auction mechanism decides commodity allocation and transaction prices when the mobile users and MCC providers submit bids and asks.

Algorithm 2 Winner Determination Algorithm of MCC Combinatorial Auction

Input:
The output of Algorithm 1
Output:
1) The allocation decision $\boldsymbol{WB}$, $\boldsymbol{WA}$
2) The vector of transaction prices $\boldsymbol{P}$.

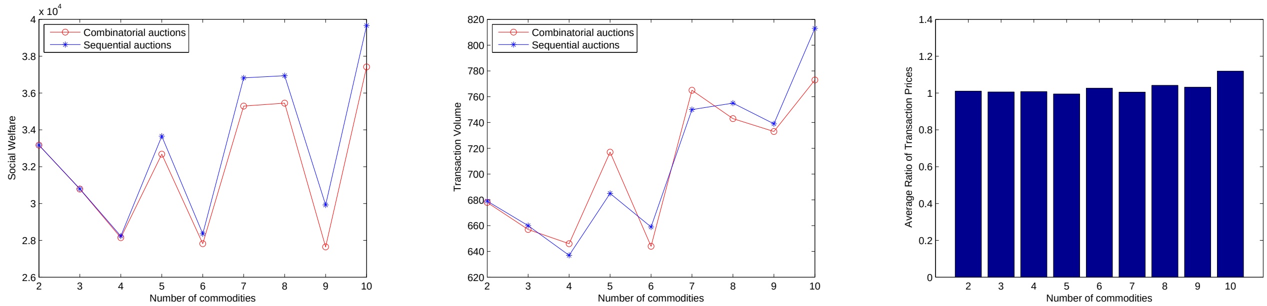
```
Construction of b, a, and get IP
{Phase I: Optimization}
Relax IP to LD by the Lagrangen Relaxation multipliers λ
Initialization k = 1, λ^(1) = (1, ..., 1), g^(1), ε > 0
while g^(k) ≥ ε do
   Compute x^(k), y^(k)
   t^(k) = (L̄ − L(x^(k), y^(k); λ^(k)))/‖ g^(k) ‖^2
   λ^(k+1) = λ^(k) + t^(k) g^(k)
   k = k + 1
end while
{Phase II: Transformation of Optimal Solution}
Remove dummy good prices from λ, P = (λ_1, ..., λ_|R|)
for r = 1 to |R| do
   With x, y, merge allocations of sub-users and sub-providers into WB, WA
end for
```

The allocation decision ($\boldsymbol{WB}$, $\boldsymbol{WA}$) and transaction prices ($\boldsymbol{P}$) are obtained by executing Algorithm 2, and these results are published on the auction platform. $\boldsymbol{WB}$ is an $|R| \times |I|$ matrix, where WB_{ri} denotes the amount of commodity r allocated to user i. Similarly, $\boldsymbol{WA}$ is an $|R| \times |J|$ matrix, where WA_{rj} denotes the quantity of good r sold by the provider j. $\boldsymbol{P}$ is a vector of $|R|$ elements, where P_r denotes the transaction price of commodity r. The platform matches users and providers according to $\boldsymbol{WB}$ and $\boldsymbol{WA}$ in sequence. Then it reports the user allocation to providers. The match result and amount that each user needs to pay are calculated and announced to users. Once the charging and payment are complete, users and providers establish the connection and providers start to give services to users.

Obviously, Algorithm 2 is **individually rational** because mobile users are never charged more than their valuations as a result of the allocation. Besides, it is **budget-balanced** because the total profits of providers is the same as the total payments by users.

VI. Simulation and Evaluation

The main objective of our auction mechanism is to allocate MCC resources effectively. Therefore, we focus on examining the allocation performance of our mechanism under illustrative mobile user's demands and provider's offer distributions. Furthermore, the computational efficiency is also an important criterion, for mechanisms should be designed to require as little computation as possible. Because the popular simulation softwares (e.g., CloudSim) support neither auction protocols nor price generation, we simulate auctions with different kinds

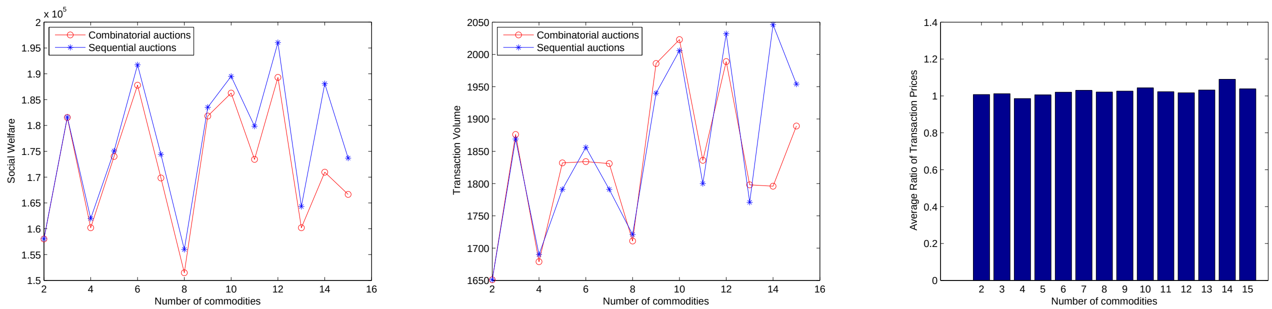


(a) Social welfare of combinatorial auctions compared with sequential auctions

(b) Transaction volume of combinatorial auctions compared with sequential auctions

(c) Ratio of transaction prices

Fig. 4. Allocation performance of the proposed MCC combinatorial double auction mechanism (Scenario 1)

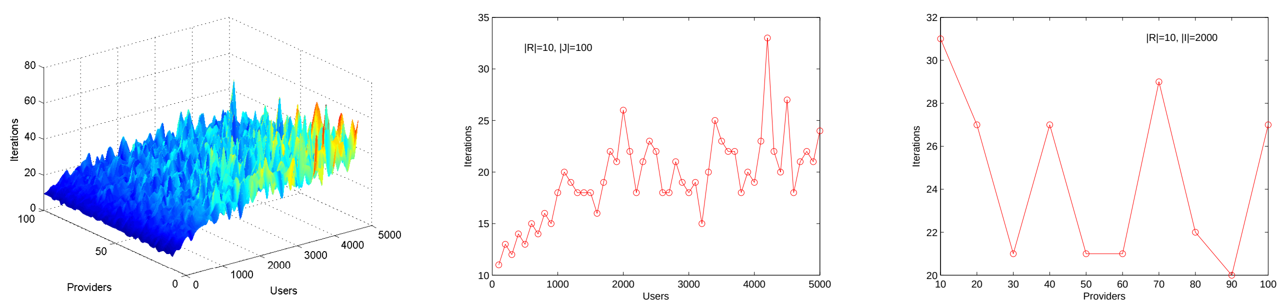


(a) Social welfare of combinatorial auctions compared with sequential auctions

(b) Transaction volume of combinatorial auctions compared with sequential auctions

(c) Ratio of transaction prices

Fig. 5. Allocation performance of the proposed MCC combinatorial double auction mechanism (Scenario 2)



(a) Iterations with different amount of users and providers

(b) Iterations with different amount of users

(c) Iterations with different amount of providers

Fig. 6. Simulation results: computational efficiency (iterations executed to get the optimal solution) of the proposed MCC combinatorial double auction mechanism

of scales to evaluate our solution in Matlab.

There are some research work applying various auction mechanisms to allocate cloud or mobile cloud resources. But we do not compare our auction mechanism with these solutions for the following reasons.

Firstly, no prior solutions have implemented online combinatorial double auctions. The existing solutions are single-sided auctions, which only support one-to-many negotiations. Usually the buyers are bidders to bid goods provided by the auctioneer. In addition, most of them are not combinatorial auctions, i.e., the bidders only bid one type of goods in one auction. Our solution enables buyers and sellers to submit bids simultaneously and allows users to bid a bundle of goods at one auction.

Secondly, the existing solutions often compare themselves with some theoretic auction mechanisms, which can get the social optimal allocation decisions but have not been applied in real markets [3] [13]. For example, Zhang *et al.* [3] compared their solution with the Vickrey-Clarke-Groves (VCG) mechanism. Although such theoretic mechanism can get the allocation decision with the optimal social welfare, it can not be followed in real-world cloud markets because it is too difficult for users to understand or the cost is too high. If social welfare of a feasible auction is close to such a theoretic optimal, it also proves the auction performance is acceptable. In this section, we compare our solution with the sequential single-minded double auctions on allocation performance.

We consider a simulation scenario with a set $\boldsymbol{R}$ of commodities, a set $\boldsymbol{I}$ of mobile users and a set $\boldsymbol{J}$ of MCC providers. Because Algorithm 1 can preprocess all the complicated origin bids and asks without loss of generality, suppose each user only submits an atomic bid ($< S, v^S >$), and each provider only offers an ask for one type of goods ($< r, c, q >$). The bundle S_i that user i bids is selected from the $2^{|\boldsymbol{R}|} - 1$ subsets of $\boldsymbol{R}$ randomly. User values, provider's offered prices and quantities are random numbers.

To compare the allocation performance of the proposed combinatorial double auction with the single-minded auctions, we can construct $|\boldsymbol{R}|$ sequential auctions $a_1, \ldots, a_r, \ldots, a_{|\boldsymbol{R}|}$, where the $r-th$ auction sells the commodity r. If user i submits $B_i = (< S_i, v_i^S >), S_i = \{r_l, r_m, r_n\}$, the bid can be divided into *3* bids for the single-minded auctions sa_l, sa_m and sa_n, and the value for each bid is set as $v_i^S/3$. We evaluate the following three criterions:

1) The **Social Welfare**, E_s, is the total payoffs/utilities of the sellers and buyers. The social welfare of the proposed combinatorial auction $E_s(CA)$ is the optimal value of objective function $Z(\boldsymbol{x}, \boldsymbol{y})$, and that of the sequential auctions $E_s(SA)$ is the sum of $|\boldsymbol{R}|$ sequential auctions: $E_s(SA) = \sum_{r \in \boldsymbol{R}} E_s(a_r)$.
2) The **Transaction Volume**, E_v, is the quantity of the goods transacted successfully, i.e., the amount of the transactions, which reflects the market efficiency of the auction mechanisms. The larger transaction volume, the better market efficiency. The transaction volume of the proposed combinatorial auction $E_v(CA) = \sum_{j \in \boldsymbol{J}} y_j$, and that of the sequential auctions $E_v(SA) = \sum_{r \in \boldsymbol{R}} E_v(a_r)$.
3) The **Average Ratio of Transaction Prices**, α, is the average ration of transaction price in our combinatorial auction to that in sequential auctions. $\alpha = (\sum_{r \in \boldsymbol{R}} P_r^{CA}/p_r^{SA})/|\boldsymbol{R}|$.

In order to provide optimal solutions to each single-minded auction a_r, we use the Marshallian path to match bids and asks [23]. The Marshallian path is simply a sequence of trades from left to right along the supply and demand curves. If the maximum valuation of a buyer is no less than the minimum cost of a seller, transaction occurs. The action is repeated until no valuation is equal or more than a cost. The Marshallian path gives a theoretic description of how to achieve the market equilibrium, therefore the obtained allocation is social optimum. The final trade of the Marshallian path is constrained to be near the competitive equilibrium, and the trade price can be deemed as the equilibrium price.

Figures 4 and 5 give the comparisons of the combinatorial auctions and sequential auction on social welfare, transaction volume and transaction price in two scenarios on different scales. The amount of users and that of providers are fixed, while the number of commodities is on the increase. In Fig. 4, $|\boldsymbol{I}| = 2000, |\boldsymbol{J}| = 100$ and $|\boldsymbol{R}| = 2, 3, \ldots 10$. While in Fig. 5, $|\boldsymbol{I}| = 5000, |\boldsymbol{J}| = 200$ and $|\boldsymbol{R}| = 2, 3, \ldots 15$.

Comparing the simulation results, we can observe that as the number of commodities increases, the allocation performance of combinatorial auctions is always close to the optimal sequential auctions. Furthermore, the transaction prices are stable.

To evaluate the computational efficiency of our mechanism, we analyze how many iterations are executed in simulation markets on different scales, as shown in Fig. 6. The number of commodities is fixed, $|\boldsymbol{R}| = 10$. In Fig. 6a, the amount of users and that of providers are both on the increase. There are *5000* users and *100* providers at most. The peak value of iterations is *59* and the mean value is *23.68*. Figure 6b gives the iteration curve with different amount of users, when the number of providers is also fixed, $|\boldsymbol{J}| = 100$. Figure 6c shows iterations varying with the amount of providers with fixed numbers of commodities and users. The results prove that the algorithm is feasible.

Overall, from the simulation results we can conclude that: first, allocation efficiency of our approach is high because the social welfare and transaction volume of our approach are close to the social optimal solution and transaction prices are stable. Second, our WDP algorithm is convergent and can obtain the optimal results with acceptable iterations.

VII. Conclusion

Auctions are gradually adopted to solve resource allocation and pricing problems in cloud computing, and many cloud auction mechanisms are designed. However these solutions are unsuitable in MCC markets. In this paper, for the first time, we apply the combinatorial double auction mechanism in MCC resource allocation and design an online auction framework to implement the mechanism. It enables mobile users to bid bundles of cloud services at one auction. For the consideration of facilitating mobile users, a novel bidding language is designed to express user's valuations concisely and it only needs to transmit few data via wireless networks. Then a model of WDP for our auction mechanism is formulated and an algorithm is designed to solve WDP, which can determine winners and prices of each auction in affordable time. The allocation obtained by our approach is individually rational and budget-balanced. At last, we design simulation scenarios to evaluate our solution. The experiment results show that the allocation performance of our solution is very close to the social optimal allocation and the computational cost is also feasible.

References

[1] R. Buyya, R. Ranjan, and R. N. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services," in *Algorithms and architectures for parallel processing*. Springer, 2010, pp. 13–31.

[2] C. Lee, P. Wang, and D. Niyato, “A real-time group auction system for efficient allocation of cloud internet applications,” 2013.

[3] H. Zhang, B. Li, H. Jiang, F. Liu, A. V. Vasilakos, and J. Liu, “A framework for truthful online auctions in cloud computing with heterogeneous user demands,” in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 1510–1518.

[4] X. Shi, K. Xu, J. Liu, and Y. Wang, “Continuous double auction mechanism and bidding strategies in cloud computing markets,” *arXiv preprint arXiv:1307.6066*, 2013.

[5] S. Zaman and D. Grosu, “Combinatorial auction-based mechanisms for vm provisioning and allocation in clouds,” in *Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on*. IEEE, 2012, pp. 729–734.

[6] S. Perez, “Mobile cloud computing: $9.5 billion by 2014,” http://www.juniperresearch.com/, 2010.

[7] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger, “Economic models for resource management and scheduling in grid computing,” *Concurrency and computation: practice and experience*, vol. 14, no. 13-15, pp. 1507–1542, 2002.

[8] S. De Vries and R. V. Vohra, “Combinatorial auctions: A survey,” *INFORMS Journal on computing*, vol. 15, no. 3, pp. 284–309, 2003.

[9] L. Y. Chu, “Truthful bundle/multiunit double auctions,” *Management Science*, vol. 55, no. 7, pp. 1184–1198, 2009.

[10] P. Vytelingum, D. Cliff, and N. R. Jennings, “Strategic bidding in continuous double auctions,” *Artificial Intelligence*, vol. 172, no. 14, pp. 1700–1729, 2008.

[11] I. SUTHERLAiND, “A futures market in computer time,” 1968.

[12] R. Wolski, J. S. Plank, J. Brevik, and T. Bryan, “Analyzing market-based resource allocation strategies for the computational grid,” *International Journal of High Performance Computing Applications*, vol. 15, no. 3, pp. 258–281, 2001.

[13] G. Vinu Prasad, S. Rao, and A. S. Prasad, “A combinatorial auction mechanism for multiple resource procurement in cloud computing.”

[14] D. Niyato, Y. Zhang, and P. Wang, “An auction mechanism for resource allocation in mobile cloud computing systems.” 2013.

[15] N. Fernando, S. W. Loke, and W. Rahayu, “Mobile cloud computing: A survey,” *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2013.

[16] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, “A performance analysis of ec2 cloud computing services for scientific computing,” in *Cloud Computing*. Springer, 2010, pp. 115–131.

[17] I. Menache, A. Ozdaglar, and N. Shimkin, “Socially optimal pricing of cloud computing resources,” in *Proceedings of the 5th International ICST Conference on Performance Evaluation Methodologies and Tools*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011, pp. 322–331.

[18] D. Lehmann, L. I. Oćallaghan, and Y. Shoham, “Truth revelation in approximately efficient combinatorial auctions,” *Journal of the ACM (JACM)*, vol. 49, no. 5, pp. 577–602, 2002.

[19] C. Boutilier and H. H. Hoos, “Bidding languages for combinatorial auctions,” in *International Joint Conference on Artificial Intelligence*, vol. 17, no. 1. LAWRENCE ERLBAUM ASSOCIATES LTD, 2001, pp. 1211–1217.

[20] T. Sandholm, “Approaches to winner determination in combinatorial auctions,” *Decision Support Systems*, vol. 28, no. 1, pp. 165–176, 2000.

[21] T. Sandholm and S. Suri, “Bob: Improved winner determination in combinatorial auctions and generalizations,” *Artificial Intelligence*, vol. 145, no. 1, pp. 33–58, 2003.

[22] F. Fumero, “A modified subgradient algorithm for lagrangean relaxation,” *Computers & Operations Research*, vol. 28, no. 1, pp. 33–52, 2001.

[23] P. J. Brewer, M. Huang, B. Nelson, and C. R. Plott, “On the behavioral foundations of the law of supply and demand: Human convergence and robot randomness,” *Experimental economics*, vol. 5, no. 3, pp. 179–208, 2002.