This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2855746, IEEE Transactions on Industrial Informatics

Double Auction-based Resource Allocation for Mobile Edge Computing in Industrial Internet of Things

Wen Sun, *Member, IEEE*, Jiajia Liu, *Senior Member, IEEE*, Yanlin Yue, *Student Member, IEEE*, Haibin Zhang, *Member, IEEE*

***Abstract*—Mobile edge computing (MEC) yields significant paradigm shift in industrial Internet of things (IIoT), by bringing resource-rich data center near to the lightweight IIoT mobile devices (MDs). In MEC, resource allocation and network economics need to be jointly addressed to maximize system efficiency and incentivize price-driven agents, whereas this joint problem is under the locality constraints, i.e., an edge server can only serve multiple IIoT MDs in the vicinity constrained by its limited computing resource. In this paper, we investigate the joint problem of network economics and resource allocation in MEC where IIoT MDs request offloading with claimed bids and edge servers provide their limited computing service with ask prices. Particularly, we propose two double auction schemes with dynamic pricing in MEC, namely a breakeven-based double auction (BDA) and a more efficient dynamic pricing based double auction (DPDA), to determine the matched pairs between IIoT MDs and edge servers, as well as the pricing mechanisms for high system efficiency, under the locality constraints. Through theoretical analysis, both algorithms are proved to be budget-balanced, individual profit, system efficient, and truthful. Extensive simulations have been conducted to evaluate the performance of the proposed algorithms and the simulation results indicate that the proposed DPDA and BDA can significantly improve the system efficiency of MEC in IIoT.**

***Index Terms*—mobile edge computing, auction, network economics, resource allocation.**

I. Introduction

Industrial Internet of things (IIoT), as a subset of IoT in industrial applications, brings together billions of devices together to monitor and analyze massive industrial data [1], [2]. IIoT boosts revenues of companies, transforms workforce, and opens up a new era of economic growth, eventually leading to *Industry 4.0* [3]. IIoT mobile devices (MDs) need to process computational-intensive tasks for intelligent decisions, which conflicts with its constrained physical size. The required computational-intensive tasks urges heavy burdens on the communication and computation techniques, and becomes bottleneck, in terms of data processing, data latency, and traffic overhead, for IIoT to evolve to be more intelligent and more powerful.

W. Sun and J. Liu are with the State Key Laboratory of Integrated Services Networks, School of Cyber Engineering, Xidian University, Xian, 710071, China (Corresponding author e-mail: liujiajia@xidian.edu.cn). Y. Yue and H. Zhang are with School of Cyber Engineering, Xidian University, Xian, 710071, China.

Mobile edge computing (MEC) [4], [5], as an emerging prominent computing paradigm, provides IIoT MDs a proximity cloud computing service through edge servers. An edge server is a mobility-enhanced and resource-rich data center, which has speedy access to the Internet and users, and offloads computing burden of mobile devices through proximity based services [6]. One distinct advantage of MEC, over the conventional cloud computing, is the reduced delay and improved reliability, as data traffic does not need to travel through the Internet to remote clouds [7], [8]. Especially for industrial applications, MEC avoids unnecessary exposure to unreliable agents, which would deteriorate the performance of IIoT [9].

Although MEC finds prominent application in IIoT, it faces unprecedented challenges, such as energy-delay constraint MEC offloading, resource allocation for various edges and MDs, and network economics for profit-driven edge servers and IIoT MDs [10]. It is also noticed that resource allocation and network economics exhibit locality in MEC where MDs can only offload to its neighboring edge servers with different preferences and an edge server can only serve its proximity MDs with limited computing resource. Moreover, in IIoT, MDs, BSs, and edge servers typically belong to different authorities, and are profit-driven caring only their own utilities. Therefore it is imperative to design an incentive framework to appropriately encourage the participation of MDs and edge servers, whereas allocating computing resource with locality-awareness to maximize the system efficiency.

There have been existing works on MEC regarding network economics [11]. The two-sided interaction where IIoT MDs request offloading with claimed bids and edge servers provide their limited computing service with ask prices, can be well modeled using a double auction mechanism. Jin *et al.* [12] proposed a feasible and truthful incentive mechanism, to coordinate the resource auction between MDs as service users (buyers) and cloudlets as service providers (sellers). However, their scheme only allows the one-to-one matching between edge server and MD. That is the computing resource of an edge server can only serve one MD at a time, which is not realistic in IIoT applications. Iosifidis *et al.* [13] assumed each mobile network operator can employ multiple APs and each AP can serve traffic from multiple operators. They designed an iterative double-auction mechanism that ensures the efficient operation of the market by maximizing the differences between the operators offloading benefits and APs offloading costs.

Note that existing works on network economics (modeling using double auction) are mostly based on breakeven-based mechanism and assume each edge server can only serve one MD, without considering locality characteristics and the limited computing resource of edge servers. To bridge the gap between the existing literature and the allocation problem, in this paper, we investigate the joint problem of network economics and resource allocation in MEC to maximize the number of successful trades. The contributions of this paper are mainly threefold as follow.

- Firstly, we model the two-sided interaction between MEC servers and IIoT MDs where IIoT MDs request computing service with claimed bids and edge servers sell their computing service with a reported ask price, under the constraints of limited computing resource of edge servers and locality between edge servers and IIoT MDs. A general double auction framework is described to address the interaction and maximize the system efficiency, while meeting the desired economic properties in terms of budge balance, truthfulness, economic efficiency, and individual rationality.
- Secondly, we propose two double auction schemes with dynamic pricing in MEC, namely a breakeven-based double auction (BDA), a more efficient dynamic pricing based double auction (DPDA). In BDA, a breakeven (threshold) index is leveraged, based on which both winner buyers and sellers are selected. BDA is proved to meet all the desired economic properties. In order to further improve the system efficiency of MEC, DPDA is proposed to keep as many feasible pairs of buyers and sellers as possible. Through theoretical analysis, DPDA can achieve a fairly high system efficiency while sacrificing truthfulness of sellers.
- Thirdly, extensive experimental simulations have been conducted to verify the performances of the proposed algorithms. Through simulation, the proposed BDA and DPDA outperform the existing works, and DPDA could get a even higher system efficiency than BDA. Therefore, in MEC when service provider could regulate the behavior of edge servers, DPDA is preferred for high system efficiency, while in the case of service provider cannot regulate the edge servers, BDA is a better choice to maintain a healthy market.

The remainder of this paper is organized as follows. Section II reviews the related works. Section III describes the system models of this work, and the problem formulation. Section IV proposes two auction schemes, i.e., BDA and DPDA. In Section V, we provide numerical results and compare the performance of the schemes. Finally, Section VI concludes the paper.

II. Related Works

The existing works on resource allocation in MEC have focused on MEC offloading (MECO) [14], [15]. Mao *et al.* [5] investigated a green MEC system with energy harvesting devices and developed a Lyapunov optimization-based dynamic computation offloading algorithm, which jointly decides the offloading decision, CPU-cycle frequencies, and transmit power. You *et al.* [16] studied resource allocation for a multi-user MECO system based on TDMA and OFDMA. In their work, the optimal resource allocation problem is formulated as a convex optimization problem for minimizing the weighted sum mobile energy consumption under the constraint on computation latency. Wang *et al.* [17] considered joint problem of MECO and interference management in the scenario of one edge server and multiple UEs. They formulated the offloading decision, resource block allocation, and the MEC resource allocation as optimization problems, where MECO decision is made according to the local computation and offloading overhead, then physical resource block (PRB) is assigned using graph coloring method. Feng *et al.* [18] proposed a framework called the autonomous vehicle edge (AVE) which is used to perform edge computing on the road to improve vehicle computing power in a decentralized manner. The AVE framework manages idle computing resources on the vehicle and leverages them to provide computing services in a dynamic vehicle environment without deploying specific infrastructure. In their later work [19], [20], a joint caching and offloading mechanism is proposed to solve the task uploading, task execution and results downloading problems in MEC.

To consider network economics in MEC, Jin *et al.* [21] extended their work in [12] and proposed another two incentive mechanism to charge MDs and reward cloudlets. They first proposed a feasible and truthful incentive mechanism, to coordinate the resource auction between mobile devices as buyers and sellers, and an efficient not not truthful mechanisms. Besides MEC, network economics have been well studied in participatory sensing and spectrum allocation [22]–[25]. Lee *et al.* [26] designed a reverse auction based dynamic price incentive scheme where contributors sell their sensing data to a service provider, and the service provider pay for the contributors to minimize and stabilize incentive cost while maintaining adequate number of participants. Yang *et al.* [27] considered a platform-centric model and a user-centric model for incentives in participator sensing, among which an auction-based incentive mechanism was designed for the user-centric model. In cooperative communication, Li *et al.* [28] focused on stimulate the relay nodes to participate in the trade.

Note that works on double auction in MEC are mostly based on breakeven-based mechanism and assume each edge server can only serve one MD. In this paper, we first consider the computing resource constraints in MEC and improve the breakeven based double auction scheme to be applied in one-to-many scenario. Then to further improve the system efficiency, DADP is proposed which dynamically assigns the computing resource of edge servers to MDs.

III. System Model

A. System Model

Fig. 1 illustrates an example of MEC in the application of IIoT where IIoT devices, such as mechanic hands, industrial camera, offload computing-intensive tasks to edge servers in the proximity, to save energy and reduce processing delay, also alleviate the pressure on the backhaul networks. For delay

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2855746, IEEE Transactions on Industrial Informatics

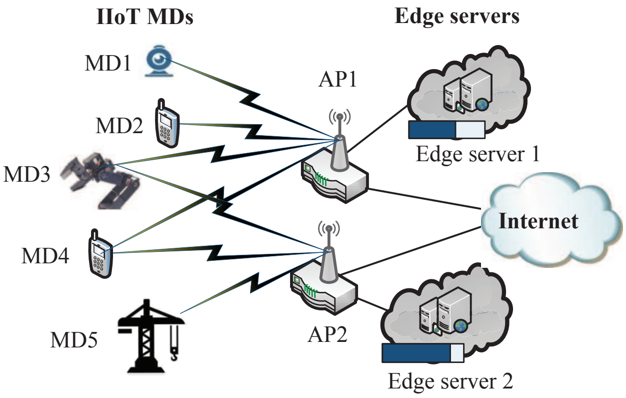


Fig. 1. Illustrating an example of MEC in the application of industry IoT. Each MD bids for computation resource from its neighboring edge servers, e.g., MD3 bids for edge server 1 and 2, while an edge server can serve its local MDs, e.g., for edge server 1 the candidate buyers are MD1, MD2, MD3, and MD4.

constraints, an IIoT device tends to connect to its neighboring edge servers.

Let $\mathcal{M} = \{1, 2, ..., M\}$ be the set of edge servers, and $M = |\mathcal{M}|$. A dynamic set of edge servers are typically lightweight, being able to deal with computation-intensive tasks, but with limited capacity. In MEC, the computing capability of each edge server is limited as well as the application sources that it can serve. Thus at the beginning of each application period, the edge server announces its computing resources, $\mathcal{R} = \{R_1, R_2, ..., R_M\}$, where R_M is the number of available resource units for M^{th} edge server and M is the number of available edge servers.

Let $\mathcal{N} = \{1, 2, ..., N\}$ be the IIoT MD set, and $N = |\mathcal{N}|$. IIoT MDs request a bundle of resources and has bid price matrix $\mathcal{B} = \{b_{i,j} : 1 \leq i \leq N, 1 \leq j \leq N\}$, where $b_{i,j}$ is to indicate the maximum price how much **it** is willing to pay for edge server j. Note that after winner determination, buyer i can only win the computing resource from seller j. In this paper b_i is used to represent the winning bid from buyer i. For each buyer, it has different valuations of the edge servers as it can attain different serving experience in terms of delay by different edge servers. In this paper, we assume the computing task is atomic and cannot be divided. The bid price MD j is willing to pay to different edge servers is different, due to his location relationship with the edge server and his/her preference. Edge server j asks a reward of a_j for one unit of service. In this case, an edge server could serve multiple users, and a MD could only be served by an edge server.

The two-sided interaction between IIoT devices and edge servers can be well-modeled as a double auction where IIoT devices are buyers, and edge servers are sellers. The service provider typically plays the role of auctioneer as a trusted third party, who determine the matching and payment scheme for both buyers and sellers, also be responsible for authentication and security check for the agents. The double auction mechanism consists of the stage of matching and pricing determination. In the stage of the matching determination, the auctioneer decides the feasible pairs of IIoT MDs and edge servers and sort them in the natural order (ordering the buyers in decreasing order of their bids and ordering the sellers in increasing order of their asks). The matching factor between buyer i and seller j is $\sigma(i) = j$. The payment scheme determines how much the auctioneer charges the IIoT MDs, and how much it pays the edge servers for processing the computing task.

Definition 1 *(Utility of IIoT MD). The utility of an IIoT MD i Π_i^b refers to the true valuation of the computing task v_i minus the price that it pays to the auctioneer p_i^b. The utility of MD i depends on its bid price b_i, the bid price of other buyers except buyer i $\mathcal{B}_{-i}$, and the asks of sellers $\mathcal{A}$, i.e.,*

$$\Pi_i^b(b_i, \mathcal{B}_{-i}, \mathcal{A}) = \begin{cases} v_i - p_i^b, & b_i \in W_b, \\ 0 & \textit{otherwise}. \end{cases} \tag{1}$$

The true valuation v_i of buyer i describes the true price that buyer i is willing to pay for the computing service. In order to avoid manipulation, the bid price shall equals to $b_i = v_i$.

Definition 2 *(Utility of Edge Server). The utility of an edge server is the payment from the auctioneer p_j^s minus its cost for processing computing tasks c_j. The utility of edge server j depends on its ask price a_i, the ask price of other buyers except seller j $\mathcal{A}_{-j}$, and the bids of buyers $\mathcal{B}$, i.e.,*

$$\Pi_j^s(a_j, \mathcal{B}, \mathcal{A}_{-j}) = \begin{cases} \sum_{b_i \in W_b} \sigma_{i,j}\left(p_j^s - c_j\right), & a_i \in W_s, \\ 0 & \textit{otherwise}. \end{cases} \tag{2}$$

The utility of the auctioneer is the difference between the total charge from the IIoT MDs and the payment to the edge servers. The utility of the auctioneer can be expressed as follows

$$\Pi^a = \sum_{b_i \in W_b} p_i^b - \sum_{a_i \in W_s} p_j^s. \tag{3}$$

The desired properties of the double auction design are as follows. (1) *Budget Balance*: A double auction scheme meets budget balance when the auctioneer does not lose money during the trade. In other words, the total price the auctioneer pays edge servers should be not less than the payment the auctioneer charges the IIoT MDs, i.e., $\sum_{b_i \in W_b} p_i^b \geq \sum_{a_j \in W_s} p_j^s$. (2) *Truthfulness*: A double auction scheme is truthful when an agent cannot improve its utility by reporting another untruthful bid $\hat{v}_i$. Particularly, if buyer i bids untruthfully with $\hat{v}_i$ for resource from seller j, its utility is less than that when it bids v_i. (3) *Economic Efficiency*: System efficiency can be measured in terms of social welfare, the number of successful trades, the revenue of edge servers, etc. In this paper we consider the objective as maximizing the number of successful trades [12]. (4) *Individual Rationality*: We assume all the entities are rational, ie., no one should lose from joining the auction. Particularly, the price p_i^b the i^{th} buyer needs to pay should be lower than its bid price b_i, i.e., $p_i^b \leq b_i$. The payment for j^{th} sell p_j^s needs to be higher than its ask s_j, i.e., $p_j^s \geq a_j$.

B. Problem Formulation

For a pair of IIoT MD and edge server, the j^{th} edge server, a set of IIoT MDs is feasible if the following two conditions are met: the demanding computation resources by the selected

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2855746, IEEE Transactions on Industrial Informatics

MDs is less than the capacity of edge server, i.e., $\sum_{b_{i,j} \in W_b} \leq R_j$; (2) all the matched bids are not less than its ask, i.e., $p_j^s \geq a_i$, $p_i^b \leq b_i$.

The objective of resource allocation in MEC is determining the set of winners, including buyers and sellers, the payment ($\mathcal{P}_b$ and $\mathcal{P}_s$ respectively), to maximize the number of matched pairs, while meet the feasibility requirement for MDs, i.e.,

$$\max \quad |W_b| \tag{4}$$
$$\text{s.t.} \quad \sum\nolimits_{b_{i,j} \in W_b} \leq R_j, \tag{5}$$
$$p_j^s \geq a_i, p_i^b \leq b_i, \tag{6}$$

Depending on the supply-demand relationship, the pricing for MDs is different and time-varying. Thus the solution to the above optimization problem is a dynamic pricing strategy. Note that in IIoT application, the number of IIoT devices could be huge, thus an edge server may receive various computing requests beyond its computing capability. In this case, the limited computing resource will be assigned to the most competitive buyers according to Eqn. (5) in the following proposed schemes.

IV. Double Auction Framework for MEC

In this section, we propose two double auction schemes to solve the resource allocation problem in MEC, i.e., a breakeven-based double auction BDA and a more efficient dynamic pricing based double auction DPDA. *Breakeven* index refers to a threshold index, above which bids can be selected successfully in the trade.

A. Breakeven Based Double Auction

In the stage of screening and candidate determination of BDA, we sort the bids and asks in natural order and determine the candidate set where feasible matching pairs are selected. The natural order is from the most competitive to the least, particularly, for asks in ascending order $\mathcal{A} = \{a_1, a_2, ..., a_M\}$, $0 \leq a_1 \leq a_2 \leq \cdots \leq a_M$, then we sort all the non-zero bids from buyers in descending order $\mathcal{B}' = \{b_{g_1,q_1}, b_{g_2,q_2}, ..., b_{g_n,q_n}\}$ where b_{g_n,q_n} refers to the bid request from buyer g_n to seller q_n, and sorted as the n^{th} element in $\mathcal{B}'$, $n = |\mathcal{B}'|$. In this case, a buyer may appear several times with respective of its bids for different sellers.

In order to reduce the computational complexity, we eliminate the failed buyers and sellers reduced by breakeven (from Line 4 to Line 7). The breakeven is determined as how many winning sellers. In order to balance between the number of selected sellers and buyers, the median of the seller set $\mathcal{A}$ is determined like [12] as a_β, where $\beta = \lfloor \frac{M+1}{2} \rfloor$. The bid threshold is the maximum index in $\mathcal{B}'$ that satisfies $\kappa = \arg\max_k \{b_{g_{k+1},q_{k+1}} < a_\beta\}$. Then all the asks that are higher than a_β are eliminated, $\mathcal{A}' \leftarrow \mathcal{A} \setminus \{a_\beta, a_{\beta+1}, ..., a_M\}$. For candidate buyers, all the buyers that are higher than $b_{g_{k+1},q_{k+1}}$ are selected, $\mathcal{B}' \leftarrow \mathcal{B}' \setminus \{b_{g_{\kappa+1},q_{\kappa+1}}, ..., b_{g_n,q_n}\}$. A pair of buyer and seller is feasible if the following two requirements are met, i.e., its bid is not less than the corresponding ask $b_{g_i,q_i} \geq a_{q_i}$ and requested seller still has computing resource to allocate $r_{q_i} > 0$. The elements that meet the first condition are added in the candidate set $\mathcal{B}_c = \{b_{g_i,q_i} | b_{g_i,q_i} \geq a_{q_i}\}$ (see Line 8 in Algorithm 1).

Then from the most competitive bid, we check its capability feasibility condition (from Line 9 to 24 in Algorithm 1) where Line 11 to 16 shows the case that the feasible capability requirement is met, and Line 17 to 24 shows the case that feasible capability requirement is not met.

- If the feasible capability requirement is met, the bidder and seller are added to the candidate set, i.e., $W_b \leftarrow W_b \cup \{b_{g_i}\}$, $W_s \leftarrow W_s \cup \{a_{q_i}\}$, and delete its bids for other edge servers from the candidate buyer set (from Line 13 to 15), as an IIoT tends to proceed its computing tasks at one edge.
- If the feasibility condition does not hold, the corresponding ask has assigned all its computing resource, and the left capability $r_{q_i} = 0$. The payment of the selected feasible buyers for seller q_i is determined the same as the highest bid who loses the trade $p_k^b \leftarrow b_{g_i,q_i}$ (Line 17 to 19 in Algorithm 1).

In the stage of winner determination and pricing, we leverage on *trade reduction* mechanism to determine the final winners from candidates as well as the payment to the winners (see Line 25 to 28 in Algorithm 1). For bids that were not eliminated due to capability feasible requirement, the winning buyers are charged with a price b_{g_κ,q_κ}. All the selected sellers $\forall a_j \in W_s$ are determined to receive the price as a_β.

Remarks: This work is different from [12] in the aspects that our work allows an edge server to serve multiple users. Moreover, we eliminate the duplicate buyers during the screening. By doing this, we can alleviate the situation that sellers in the backward order elimiate the candidate buyers during *buyer elimination*, and loose the chance to incorporate other potential buyers.

B. Dynamic Pricing Based Double Auction

The system efficiency of BDA is limited, as lots of feasible pair of buyer and seller are not included. The basic idea of DPDA is to keep as many feasible pairs of matchings as possible, and provide truthful pricing scheme. Algorithm 2 shows the details of DPDA.

The winner determination is similar to that of BDA. Buyers and sellers are first sorted in natural order, as shown in Line 2 and 3 in Algorithm 2. Then the feasible pairs of buyers and sellers are added in W_b and W_s. The first feasible requirement is slighted changed to $b_{g_i,q_i} \geq a_{q_i+1}$ in order to appropriately pricing all the winning buyers. Particularly, a pair of buyer and seller is feasible if the following two requirements are met, i.e., its bid is not less than the next of its corresponding ask $b_{g_i,q_i} \geq a_{q_i+1}$ and requested seller still has computing resource to allocate $r_{q_i} > 0$. The elements that meet the first condition are added in the candidate set $\mathcal{B}_c = \{b_{g_i,q_i} | b_{g_i,q_i} \geq a_{q_i+1}\}$. The actions (from Line 6 to Line 19 in Algorithm 2) is similar to that of Algorithm 1. In the stage of pricing determination, for seller j, its winning buyers $\forall b_{i,j} \in \mathcal{B}'$ need to pay a_{j+1}. The payment for the j^{th} seller is the ask of the $(j+1)^{th}$ seller.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2855746, IEEE Transactions on Industrial Informatics

Algorithm 1 Breakeven based double auction, i.e., BDA($\mathcal{B}, \mathcal{A}, \mathcal{R}$).

Input: $\mathcal{B}$, $\mathcal{A}$, $\mathcal{R}$
Output: W_b, W_s, $\mathcal{P}_b$, $\mathcal{P}_s$, σ

Initialize $W_b \leftarrow \emptyset$, $W_s \leftarrow \emptyset$, $\mathcal{P}_b \leftarrow \emptyset$, $\mathcal{P}_s \leftarrow \emptyset$
Sort the asks of sellers in $\mathcal{A}$ in ascending order, i.e., $\mathcal{A} = \{a_1, a_2, ..., a_M\}$, $0 \le a_1 \le a_2 \le \cdots \le a_M$
Sort all the non-zero elements of $\mathcal{B}$ in descending order and form $\mathcal{B}' = \{b_{g_1,q_1}, b_{g_2,q_2}, ..., b_{g_n,q_n}\}$, $b_{g_1,q_1} \ge b_{g_2,q_2} \ge ...b_{g_n,q_n}$, $n = |\mathcal{B}'|$
Find the median of $\mathcal{A}$ as a_β where $\beta = \left\lfloor \frac{M+1}{2} \right\rfloor$
Find the bid breakeven index, $\kappa = \arg\max_k \left\{ b_{g_{k+1},q_{k+1}} < a_\beta \right\}$
Select the asks of sellers less than the ask breakeven $\mathcal{A}' \leftarrow \mathcal{A} \setminus \{a_\beta, a_{\beta+1}, ..., a_M\}$
Select the bids of buyers higher than the bid breakeven $\mathcal{B}' \leftarrow \mathcal{B}' \setminus \left\{ b_{g_{\kappa+1},q_{\kappa+1}}, ..., b_{g_n,q_n} \right\}$
$\mathcal{B}_c = \left\{ b_{g_i,q_i} | b_{g_i,q_i} \ge a_{q_i} \right\}$, $n = |\mathcal{B}_c|$
for $i = 1$ to n **do**
  if $r_{q_i} > 0$ AND $q_i < \beta$ **then**
    $W_b \leftarrow W_b \cup \{b_{g_i}\}$, $W_s \leftarrow W_s \cup \{a_{q_i}\}$
    $\sigma(i) = q_i$, $r_{q_i} \leftarrow r_{q_i} - 1$
    while $b_{g_i,q_k} \in \mathcal{B}_c$ **do**
      $\mathcal{B}_c \leftarrow \mathcal{B}_c \setminus \{b_{g_i,q_k}\}$, $n \leftarrow n - 1$
    end while
  else
    if $r_{q_i} \le 0$ **then**
      $\mathcal{B}_c \leftarrow \mathcal{B}_c \setminus \{b_{g_i,q_i}\}$, $n \leftarrow n - 1$
      while $b_{g_k,q_i} \in \mathcal{B}_c, \forall k$ **do**
        $p_k^b \leftarrow b_{g_i,q_i}$, $\mathcal{P}_b \leftarrow \{p_k^b\}$
      end while
    end if
  end if
end for
while $b_{g_i,q_i} \in \mathcal{B}_c$, $p_i^b = 0$ **do**
  $p_i^b \leftarrow b_{g_\kappa}, q_\kappa$, $\mathcal{P}_b \leftarrow \{p_i^b\}$
end while
$p_j^s \leftarrow a_\beta$, $\forall a_j \in W_s$, $\mathcal{P}_s \leftarrow \{p_j^s\}$
return W_b, W_s, $\mathcal{P}_b$, $\mathcal{P}_s$, σ

Remarks: In DPDA, a_{q_i+1} is leveraged to ensure feasibility instead of a_{q_i}. This measurement would slightly reduce the efficiency as the number of feasible buyers $b_{g_i,q_i} \ge a_{q_i+1}$ is typically smaller than that of $b_{g_i,q_i} \ge a_{q_i}$ with $a_{q_i} \le a_{q_i+1}$. Here, we slightly reduce the number of matched pairs to trade for truthfulness, as in this case a_{q_i+1} is a critical bid for all buyers (see Theorem 3).

C. A Walk-Through Example

In order to clearly understand the procedure of BDA and DPDA, as well as compare with the existing works ICAM, we give a walk-through example in the case of 5 sellers and 4 buyers. In Table I, we show bid matrix of buyers, and the asks and resources of sellers and the winner determination results are given in Table II.

Algorithm 2 Dynamic Price based Double Auction in MEC, i.e., DPDA($\mathcal{B}, \mathcal{A}, \mathcal{R}$).

Input: $\mathcal{B}$, $\mathcal{A}$, $\mathcal{R}$
Output: W_b, W_s, $\mathcal{P}_b$, $\mathcal{P}_s$, σ

// Winner determination
Sort the asks of sellers in $\mathcal{A}$ in ascending order, i.e., $\mathcal{A} = \{a_1, a_2, ..., a_M\}$, $0 \le a_1 \le a_2 \le \cdots \le a_M$
Sort all the non-zero elements of $\mathcal{B}$ in descending order and form $\mathcal{B}' = \{b_{g_1,q_1}, b_{g_2,q_2}, ..., b_{g_n,q_n}\}$ where $b_{g_1,q_1} \ge b_{g_2,q_2} \ge ...b_{g_n,q_n}$, $n = |\mathcal{B}'|$
$\mathcal{B}_c = \left\{ b_{g_i,q_i} | b_{g_i,q_i} \ge a_{q_i+1} \right\}$, $n = |\mathcal{B}_c|$
for $i = 1$ to n **do**
  if $r_{q_i} > 0$ **then**
    $W_b \leftarrow W_b \cup \{b_{g_i}\}$, $W_s \leftarrow W_s \cup \{a_{q_i}\}$
    $\sigma(i) = q_i$, $r_{q_i} \leftarrow r_{q_i} - 1$
    while $b_{g_i,q_k} \in \mathcal{B}_c$ **do**
      $\mathcal{B}_c \leftarrow \mathcal{B}_c \setminus \{b_{g_i,q_k}\}$, $n \leftarrow n - 1$
    end while
  else
    $\mathcal{B}_c \leftarrow \mathcal{B}_c \setminus \{b_{g_i,q_i}\}$, $n \leftarrow n - 1$
  end if
end for
// Pricing Determination
for $j = 1$ to $M - 1$ **do**
  $p_j^s \leftarrow a_{j+1}$, $\mathcal{P}_s \leftarrow \mathcal{P}_s \cup \{p_j^s\}$
  $p_i^b \leftarrow a_{j+1}$, $\forall b_{i,j} \in \mathcal{B}_c$
  $\mathcal{P}_b \leftarrow \mathcal{P}_b \cup \{p_i^b\}$
end for
return W_b, W_s, $\mathcal{P}_b$, $\mathcal{P}_s$, σ

For BDA, we first sort the asks of sellers and buyers in the correct order, and find the median of $\mathcal{A}$ as α_β, $\alpha_\beta = a_3 = 4$, and the maximum value less than α_β from $\mathcal{B}'$, i.e., $b_{g\kappa+1,q\kappa+1} = b_{4,2} = 3$, $\kappa = 6$. Then we remove the elements greater than or equal to α_β in $\mathcal{A}$ and remove the elements in $\mathcal{B}'$ that are less than or equal to $b_{4,2}$. For any bid $b_{g_i,q_i} \in \mathcal{B}'$, if $a_{q_i} \in \mathcal{A}'$, then $b_{p_i,q_i} \in \mathcal{B}_c$. $\mathcal{B}_c = \{b_{2,3} = 10, b_{2,1} = 8, b_{1,3} = 7, b_{4,3} = 6, b_{3,1} = 5\}$. In the stage of winner determination and pricing, we assign resources from the first element in $\mathcal{B}_c$, i.e., $b_{2,3}$. $r_3 = 2 > 0$, thus buyer 2 can achieve the service of seller 3. $W_b = W_b \cup \{2\} = \{2\}$, $W_s = W_s \cup \{3\} = \{3\}$, and then we delete the elements associated with buyer 2 in $\mathcal{B}_c$. All the sellers in W_s receive the payment α_β for every winning buyer, and the buyers that have not been charged yet in W_b have to pay $a_{g\kappa,q\kappa}$, i.e., $a_{g\kappa,q\kappa} = b_{3,1} = 5$. The remaining elements in $\mathcal{B}_c$ are the final winning bids.

For DPDA, buyers and sellers are first sorted in natural order, just like BDA, i.e., $\mathcal{A} = \{a_1 = 2, a_2 = 3, a_3 = 4, a_4 = 5, a_5 = 6\}$, $\mathcal{B}' = \{b_{2,3} = 10, b_{3,2} = 10, b_{2,1} = 8, b_{1,3} = 7, b_{4,3} = 6, b_{3,1} = 5, b_{4,2} = 3, b_{4,5} = 3, b_{1,4} = 1\}$. However, in DPDA, we do not apply the median ask as breakeven. We match the bids and asks that meet the feasibility requirement: $b_{g_i,q_i} \ge a_{q_{i+1}}$. For the first element in $\mathcal{B}'$, i.e., $b_{2,3} = 10$, the buyer 3 has an ask of 2 and the smallest number in $\mathcal{A}$ that is larger than 2 is 3. Since $b_{2,3} = 10 > 3$, $b_{2,3}$ is added to $\mathcal{B}_c$. And so forth, the candidate set $\mathcal{B}_c = \{b_{2,3} = 10, b_{3,2} =$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2855746, IEEE Transactions on Industrial Informatics

TABLE I
AN EXPLANATORY EXAMPLE

Bid matrix of 4 buyers

	s_1	s_2	s_3	s_4	s_5
b_1	0	0	7	1	0
b_2	8	0	10	0	0
b_3	5	10	0	0	0
b_4	0	3	6	0	3

(b) Ask vector of 5 sellers

Seller	s_1	s_2	s_3	s_4	s_5
Ask	3	4	2	6	5
Resource	4	3	2	2	3

$10, b_{2,1} = 8, b_{1,3} = 7, b_{4,3} = 6, b_{3,1} = 5\}$. Then we judge in order whether the element $b_{gi,qi}$ in $\mathcal{B}_c$ satisfies the condition of $r_{q_i} > 0$. For $b_{2,3}$, since $r_3 = 2 > 0$, buyer 2 and seller 3 are added to W_b and W_s respectively. $b_{2,1}$ is removed form $\mathcal{B}_c$. The price of buyer 2 is the same as the payment of seller 3, i.e. the smallest ask that is larger than the ask of seller 3. $p_2^b = p_3^s = 3$. The rest elements in $\mathcal{B}_c$ will be done in the same manner. And finally, we get $W_b = \{2, 3, 1\}$, $W_s = \{3, 2\}$, $\mathcal{P}_s = \{p_3^s = 3, p_2^s = 5, p_3^s = 3\}$, $\mathcal{P}_b = \{p_2^b = 3, p_3^b = 5, p_1^b = 3\}$, $\sigma = \{(2 \to 3), (3 \to 2), (1 \to 3)\}$.

The ICAM scheme in [12] is a one-to-one service model where one edge server can only serve one MD and one MD can only bid on one edge server. ICAM also chose the median of asks as the breakeven and payment to winning sellers. If there are multiple buyers bidding to the same seller, the maximum bid value among the losers is taken as the price of the buyer who won the seller's service. If only one buyer bids and wins the service, then the minimum bid value greater than or equal to the breakeven is taken as the price of the buyer. Therefore, $W_b = \{2, 3\}$, $W_s = \{3, 1\}$, $\mathcal{P}_s = \{p_3^s = 4, p_1^s = 4\}$, $\mathcal{P}_b = \{p_2^b = 7, p_3^b = 5\}$, $\sigma = \{(2, 3), (3, 1)\}$.

In summary, in the BDA scheme, three MDs are served. In DPDA, 4 MDs are served and in ICAM, only 2 MDs are served. The results of the example are shown in Table II.

D. Economic Properties

Lemma 1 *Both proposed double auction schemes are computational-efficient, and the computational complexity of BDA and DPDA are the same as $\mathcal{O}\left(M^2N^2\right)$.*

Proof: For BDA, as shown in Algorithm 1, for Line 2, the computational complexity of quick sort in the worst case is $\mathcal{O}(M \log M)$. Similarly as the maximum length of $\mathcal{B}'$ is MN, the computational complexity of Line 3 is $\mathcal{O}(MN \log(MN))$. The computational complexity from Line 9 to 24 is $\mathcal{O}\left(M^2N^2\right)$. The computational complexity of Algorithm 1 is $\mathcal{O}\left(M^2N^2\right)$. Similarly, for DPDA, as shown in Algorithm 2, the computational complexity of quick sort in the worst case for Line 2 and 3 is $\mathcal{O}(M \log M)$ and $\mathcal{O}(MN \log(MN))$ respectively. Then from Line 5 to 17, the computational complexity is $\mathcal{O}\left(M^2N^2\right)$. After winner determination, the maximum length of $\mathcal{B}_c$ is still MN. Then from Line 17 to 21, it takes $\mathcal{O}((M-1)MN)$. The computational complexity of Algorithm 2 is $\mathcal{O}\left(M^2N^2\right)$. ■

Theorem 1 *Both BDA and DPDA mechanisms satisfy individual rationality.*

Proof: Recall that individual rationality refers to that no one should lose from joining the auction. Particularly, the price p_i^b the i^{th} buyer needs to pay should be lower than its bid price b_i, i.e., $p_i^b \le b_i$. The payment for j^{th} sell p_j^s needs to be higher than its ask s_j, i.e., $p_j^s \ge a_j$. For BDA, it meets the requirement of individual rationality when the utility for each agent is non-negative. If a bid or ask fails in the auction, $b_i \notin W_b$, $a_j \notin W_s$, the utility of the bid or ask is zero. For seller j, its received payment $p_j^s = a_\beta$ and its asks is a_j. According to Line 10 in Algorithm 1, the necessary condition for $a_j \in W_s$ is $j < \beta$, then $a_\beta \ge a_j$ due to monotonic sorting. When buyer i wins an auction, it falls into two cases according to the left capability of its corresponding edge server. (1) When its corresponding edge server is fully assigned $r_j \le 0$, the highest bid that loses the trade is charged from buyer i (Line 20 in Algorithm 1). As buyer i wins the trade, its bid is surely higher than the charged price. (2) Otherwise, buyer i is charged the breakeven price $b_{g_{\kappa+1}, q_{\kappa+1}}$, which is lower than its bid price. Thus both buyers and sellers in BDA satisfy individual rationality. For DPDA, we only match the bids and asks that meet the feasibility requirement, i.e., $b_{i,j} \ge a_{j+1}$. Also according to the pricing stage (Line 4 18 19 of Algorithm 2), the auctioneer charges an IIoT MD and pays the edge server the same price of a_{j+1}. Thus for a buyer, its utility equals the true valuation minus its payment, $b_i - a_{j+1} \ge 0$. Similarly, for seller j, the payment it receives is higher than its ask $a_{j+1} - a_j \ge 0$. ■

Lemma 2 *Prices are monotonic if $p_{i'}^b \ge p_i^b$ for all $b_{i'} \ge b_i$, and $p_{i',j'}^s \ge p_{i,j}^s$ for all $a_{j'} \le a_j$.*

The monotonicity of the proposed auction is intuitive as bidding a smaller value will make the user backwards in the sorting. If bid or ask in the winner set, the pricing scheme must be independent of the reported bids or asks.

Theorem 2 *The proposed BDA is truthful for both buyers and sellers.*

Proof: According to [29], a bid monotonic auction is truthful if and only if it always charges critical bids from winning buyers and pays critical asks to winning sellers. For winning buyer i, p_i^b is critical if buyer i wins by submitting $b_i > p_i^b$ and loses by submitting $b_i < p_i^b$, given others' submission remain unchanged. Similarly, for winning seller j, p_j^s is critical if seller j wins by submitting $a_i < p_j^s$ and loses by submitting $a_i > p_j^s$, given others' submission remain unchanged.

For sellers, a winning seller will remain in the winning list when $a_j < a_\beta$. When seller j asks $a_j > a_\beta$, it will be removed from the winning set W_s according to BDA. Thus a_β is the critical ask of all sellers. There are two cases for the critical bids of buyers.

- When its corresponding edge server is fully assigned $r_j \le 0$, the highest bid that loses the trade (denoted as b_k) is

TABLE II
THE RESULTS OF THE EXAMPLE

	Set of winning buyers	Set of winning sellers	Successfully matched pairs
$ICAM$	$\{2,3\}$	$\{3,1\}$	$\{(2,3),(3,1)\}$
BDA	$\{2,1,3\}$	$\{3,1\}$	$\{(2,3),(1,3),(3,1)\}$
$DPDA$	$\{2,3,1,4\}$	$\{3,2,5\}$	$\{(2,3),(3,2),(1,3),(4,5)\}$

charged from buyer i (Line 20 in Algorithm 1). In this case, the highest bid that loses the trade at edge server j is the critical bid for winning buyers at seller j. For a winning buyer i at seller j, it will be in the set of top buyers within the computing capability of edge server j, thus it will be selected as winner buyers for this seller. When buyer i bids $b_i < b_k$, it will be sorted after b_k. Then b_k will be selected instead of b_i.

- Otherwise, buyer i is charged the breakeven price $b_{g_{\kappa+1},q_{\kappa+1}}$, which is the critical bid of all winning buyer in this case. For a winning buyer i, it will remain in the set of top κ buyers if its bid $b_{i,j} > b_{g_{\kappa+1},q_{\kappa+1}}$. In this case, buyer i is still in the winning set. On the other hand, when buyer i bids $b_i < b_{g_{\kappa+1},q_{\kappa+1}}$, and is sorted backward and buyer i will no longer be selected in the winner set.

Therefore, buyers are charged with their critical bids and sellers receive their critical asks in BDA. Along with the monotony, we prove the proposed BDA is truthful for both buyers and sellers, and thus economy robustness from manipulation. ∎

Theorem 3 *The proposed DPDA is truthful for buyers.*

Proof: In the stage of DPDA pricing determination, for seller j, its winning buyers $\forall b_{i,j} \in W_b$ need to pay a_{j+1}. The payment for the j^{th} seller is the ask of the $(j+1)^{th}$ seller. Here a_{j+1} is the critical bid for winning buyers at seller j. For a winning buyer i, if it changes its bid to be $b_i < a_{j+1}$, it will not be selected in W_b according to Line 4 in Algorithm 2. As long as it bids $b_i \geq a_{j+1}$, buyer i will be selected as winners. Therefore, the price charged from each buyer is the critical bid in DPDA. Together with its monotonicity, DPDA is truthful for buyers. ∎

In comparison, BDA achieves truthfulness for both buyers and sellers, but the system efficiency is limited by the breakeven-based winner determination mechanism. DPDA has higher system efficiency than BDA at the cost of truthfulness of sellers, as one seller can serve multiple buyers and all feasible sellers are selected without a breakeven. Fortunately, in MEC, it is easier for a service provider to regulate the behavior of edge servers rather than mobile users. For instance, service provider can find the truthful value of edge servers through their historical asks and behavior. Therefore, in MEC when service provider could regulate the behavior of edge servers, DPDA is preferred for high system efficiency, while in the case of service provider cannot regulate the edge servers, BDA is a better choice to maintain a healthy market.

V. PERFORMANCE EVALUATION

A. Simulation Settings

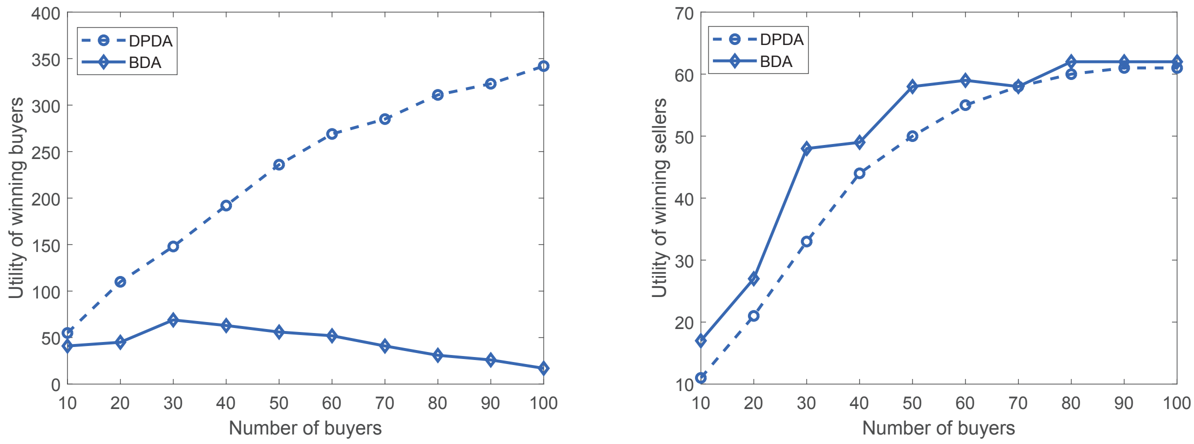
In application area with dimensions of $500 \times 500\ m^2$. 100 users are assumed to be present in the application area. Users bid for the resources (j^{th} user bid price is b^j) where the ask bid prices for users are randomly generated following a uniform distribution within the range of $[0, 14]$ without otherwise specified. Similarly, edge servers sell their computing resource with ask prices set between $[3, 10]$. The computing capacity of the MEC server between 240 MHz and 0.8 GHz. The size of requested computation data from intelligent agents is 80 MB. The number of CPU cycles required by a computation task is set between 100 Megacycles and 1 Gigacycles. The number of available channels for a BS is about 100, which means at most 100 MDs connected with the BS at the same time constrained by communication capabilities. The performance of the proposed BDA and DPDA is compared with ICAM [12].

B. Utility and System Efficiency

Fig. 2 compares the utility of buyers and sellers using DPDA and BDA over varying number of IIoT MDs. As can be seen from Fig. 2(a), the total utility of IIoT MDs using DPDA is much higher than that using BDA because DPDA significantly increases the number of successfully matched trades. It is interesting to notice that with BDA, the utility of the winning buyers increases sharply when the number of buyers is less than 30 and increases very slowly even has a tendency to decline when the number of buyers is large. This can be interpreted by the breakeven and pricing mechanism, as $a_j > a_\beta$ are eliminated, at most $M/2$ sellers are selected with quite limited computing resource capability, thus a large amount of feasible pair of trades are eliminated in BDA. In addition, as the number of buyers increases, each buyer faces greater competition and the price each winner pays will be increase, so the utility of each winner will decrease. However, the total number of successful trades does not increase too much, resulting in a decrease in total utility. Fig. 2(b) compares the utility of edge servers using DPDA and BDA over varying number of IIoT MDs. As can be seen from Fig. 2(b), the utility of edge servers using BDA is slightly higher than DPDA as the pricing scheme determines payment price slightly higher than the asks.

Fig. 3 illustrates the comparison of different algorithms in terms of the number of successful trades. Fig. 3(a) depicts the number of successful matching using BDA and DPDA over varying number of buyers. As can be seen from Fig. 3(a), the number of the successful trades of DPDA is typically higher than BDA, the advantage of DPDA is even more obvious when the number of buyers increases. ICAM is not compared in

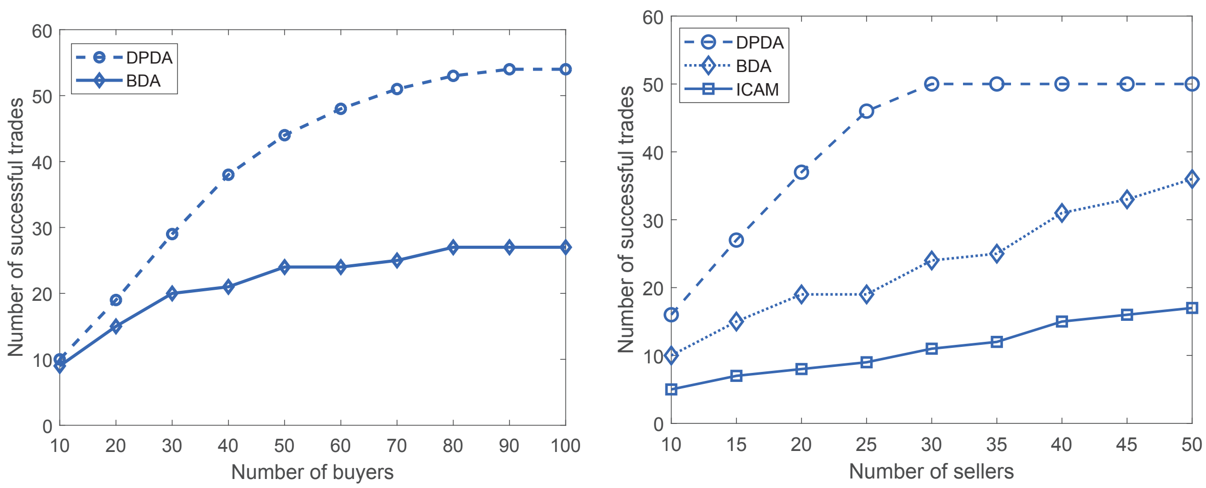
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2855746, IEEE Transactions on Industrial Informatics



(a) Utility of buyers.

(b) Utility of sellers

Fig. 2. The utility of buyers and sellers using BDA and DPDA. The utility of winning buyers or sellers is the sum of utility of all winning buyers or sellers. The bids from buyers are set between $[0, 14]$, and the asks from sellers are between $[3, 10]$.



(a) Comparison over different number of buyers

(b) Comparison over different number of sellers

Fig. 3. Comparison of different algorithms in terms of the number of successful trades.

Fig. 3(a) as the number of buyers will be bounded by the seller number and will be a constant (5 in this simulation setting) in the figure. As can be seen from Fig. 3(b), DPDA and BDA obviously outperforms ICAM as ICAM only allows an edge server serves an IIoT MD. The performance of DPDA is higher than BDA due to the similar reason that DPDA did not use a breakeven mechanism to eliminate feasible pairs of trades.

C. Individual Rationality and Budget Balance

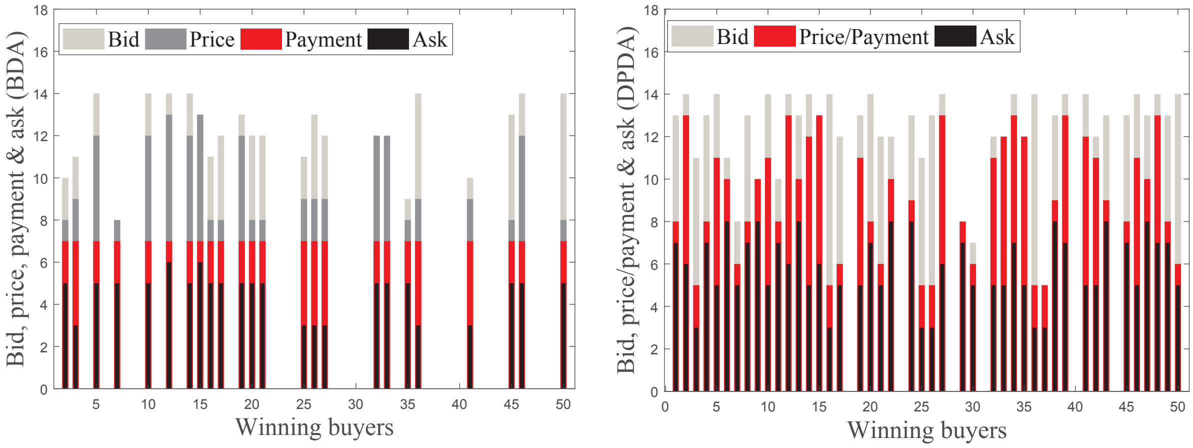
In Fig. 4(a) and 4(b), we show the bids, pricing, and asks of BDA and DPDA over 50 buyer-seller pairs. Each bar in the histogram represents the successfully matched pairs of pricing. As in DPDA there are more buyer-seller pairs successfully matched in the trade than that of BDA, the bars of DPDA histogram is denser than that of BDA. In BDA, the price charged from each winning buyer differentiates from the payment for each winning seller, thus price and payment are shown separately in Fig. 4(a). In DPDA, the price charged to each winning buyer equals to the payment rewarded to each winning seller, the pricing here presents both price and payment. As can be seen, for both BDA and DPDA, each winning buyer is charged a price not higher than its bid, while each winning seller receives a payment not less than its ask. Therefore, both BDA and DPDA are individually rational. The means that agents have sufficient incentive to participate the trade.

Fig. 5 shows the truthfulness of buyers and sellers in BDA and DPDA. As can be from Fig. 5(a), in BDA, when a buyer bids lower than its critical value, it cannot win the trade and the utility is zero. When it bids higher than the critical value, the utility is a constant, not greater than the critical value. Similarly, the behaviors of sellers in Fig. 5(b) and buyers in Fig. 5(c) can be explained.

D. Discussions

Although this paper is on MEC in IIoT, the designed algorithms can be easily applied in other MEC systems such

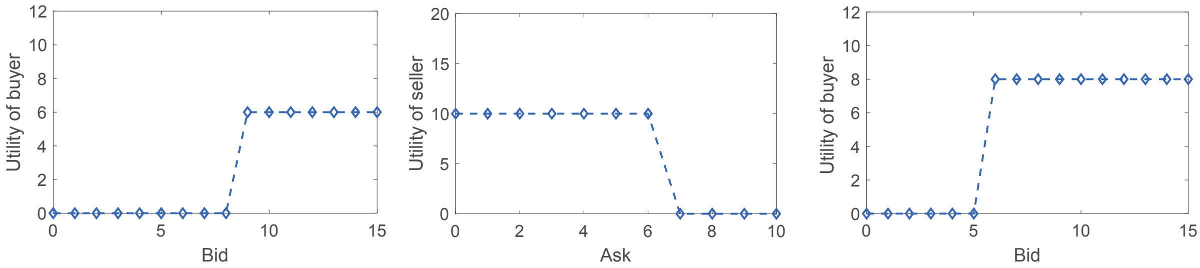
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2855746, IEEE Transactions on Industrial Informatics



(a) BDA

(b) DPDA

Fig. 4. The comparison of bids, price and ask for BDA and DPDA where price refers to the amount charged from a buyer and payment is the amount of reward received by a seller.



(a) Buyer in BDA

(b) Seller in BDA

(c) Buyer in DPDA

Fig. 5. The truthfulness of buyer and seller.

as VANET and other MEC scenario, as long as the users and edge servers are profit-driven with locality awareness. When considering user mobility, the spatio-temporal relationship between IIoT MDs and edge servers will change dynamically. For example, a UE at the service range of an edge server may move out of the service range of the current edge server, and will suffer longer delay to still offload to the current edge server, thus the offloading decision needs to be updated together with the locality relationship. We assume the locality relationship between users and edge servers remain unchanged within a time slot. In the proposed schemes, the bid matrix of buyers, reflecting the spatio-temporal relationship between MDs and edge servers, needs to be updated each time slot. Then after updating bid matrix, the proposed BDA and DPDA can be applied to mobility case easily.

In the future work, multiple computing applications on MEC servers will be considered, and each MD may simultaneously offload different types of tasks to an edge server while edge server differentiates in dealing types of tasks from MDs. Thus the heterogeneity of edge servers will be considered in terms of not only computing capability, but also the serving computing applications. In particular, in the resource allocation strategy, the feasibility requirements will be totally different as well as the bid matric. In the follow-up work, incentives will be also employed to encourage continuous participation from MDs and edge servers.

VI. Conclusions

In this paper, we have investigated the joint problem of network economics and resource allocation in MEC where an edge server can serve multiple IIoT MDs in the vicinity. Two double auction schemes were proposed with dynamic pricing in MEC, namely BDA and DPDA, which are both proved to meet the economic properties. Extensive simulations have been conducted to evaluate the performance of the proposed algorithms and the simulation results indicate that the performance of DPDA and BDA can significantly improve the system efficiency of MEC.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (61601357, 61771374, and 61771373), in part by China 111 Project (B16037), and in part by the Fundamental Research Fund for the Central Universities (JB181506, JB171501, JB181507, and JB181508).

References

[1] Z. Meng, Z. Wu, C. Muvianto, and J. Gray, “A data-oriented m2m messaging mechanism for industrial iot applications,” *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 236–246, 2017.

[2] L. Da Xu, W. He, and S. Li, “Internet of things in industries: A survey,” *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2855746, IEEE Transactions on Industrial Informatics

[3] L. Shu, L. Wang, J. Niu, C. Zhu, and M. Mukherjee, "Releasing network isolation problem in group-based industrial wireless sensor networks," *IEEE Systems Journal*, vol. 11, no. 3, pp. 1340–1350, 2015.

[4] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computingła key technology towards 5g," *ETSI white paper*, vol. 11, no. 11, pp. 1–16, 2015.

[5] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.

[6] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

[7] C. Zhu, H. Wang, X. Liu, L. Shu, L. T. Yang, and V. C. M. Leung, "A novel sensory data processing framework to integrate sensor networks with mobile cloud," *IEEE Systems Journal*, vol. 10, no. 3, pp. 1125–1136, 2016.

[8] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, 2012.

[9] C. Zhu, L. Liu, J. Jiang, L. Shu, and G. Hancke, "Analysis of energy-efficient connected target coverage algorithms for industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 135–143, 2017.

[10] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[11] A. Kiani and N. Ansari, "Toward hierarchical mobile edge computing: An auction-based profit maximization approach," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2082–2091, 2017.

[12] A.-L. Jin, W. Song, and W. Zhuang, "Auction-based resource allocation for sharing cloudlets in mobile cloud computing," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 1, pp. 45–57, 2015.

[13] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "A double-auction mechanism for mobile data-offloading markets," *IEEE/ACM Transactions on Networking (TON)*, vol. 23, no. 5, pp. 1634–1647, 2015.

[14] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, "Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[15] L. Shu, M. Hauswirth, H. Chao, M. Chen, and Y. Zhang, "Nettopo: A framework of simulation and visualization for wireless sensor networks," *Ad Hoc Netowrks*, vol. 9, no. 5, pp. 799–820, 2011.

[16] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1397–1411, 2017.

[17] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7432–7445, 2017.

[18] J. Feng, Z. Liu, C. Wu, and Y. Ji, "Ave: Autonomous vehicular edge computing framework with aco-based scheduling," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 10 660–10 675, 2017.

[19] J. Guo, Z. Song, Y. Cui, Z. Liu, and Y. Ji, "Energy-efficient resource allocation for multi-user mobile edge computing," in *IEEE GlobeCom*, 2017, pp. 1–7.

[20] Y. Cui, W. He, C. Ni, C. Guo, and Z. Liu, "Energy-efficient resource allocation for cache-assisted mobile edge computing," in *IEEE ICN*, 2017, pp. 640–648.

[21] A.-L. Jin, W. Song, P. Wang, D. Niyato, and P. Ju, "Auction mechanisms toward efficient resource sharing for cloudlets in mobile cloud computing," *IEEE Transactions on Services Computing*, vol. 9, no. 6, pp. 895–909, 2016.

[22] H. Saito and S. Shioda, "Parameter estimation method for time-variant target object using randomly deployed sensors and its application to participatory sensing," *IEEE Transactions on Mobile Computing*, vol. 14, no. 6, pp. 1259–1271, 2015.

[23] P. Zhou, Y. Zheng, and M. Li, "How long to wait? predicting bus arrival time with mobile phone based participatory sensing," *IEEE Transactions on Mobile Computing*, vol. 13, no. 6, pp. 1228–1241, 2014.

[24] T. T. Huu and C.-K. Tham, "An auction-based resource allocation model for green cloud computing," in *Proc. IEEE IC2E'13*, Santa Clara, USA, Jun. 2013.

[25] W. Wang, B. Liang, and B. Li, "Designing truthful spectrum double auctions with local markets," *IEEE Transactions on Mobile Computing*, vol. 13, no. 1, pp. 75–88, 2014.

[26] J. Lee and B. Hoh, "Sell your experiences: a market mechanism based incentive for participatory sensing," in *Proc. IEEE PerCom'12*, Manheim, Germany, Apr. 2010.

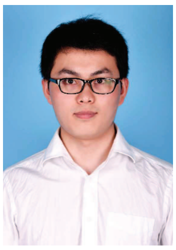
[27] D. Yang, G. Xue, and X. Fang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *Proc. ACM MobiCom'12*, Istanbul, Turkey, Aug. 2012.

[28] Y. Li, C. Liao, Y. Wang, and C. Wang, "Energy-efficient optimal relay selection in cooperative cellular networks based on double auction," *IEEE Transactions on Wireless Communications*, vol. 14, no. 8, pp. 4093–4104, 2015.

[29] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic game theory*. Cambridge University Press Cambridge, 2007, vol. 1.

Wen Sun (M'16-SM'11) is currently an associate professor with the School of Cyber Engineering, Xidian University. She received her Ph.D. degree in Electrical and Computer Engineering from National University of Singapore in 2014, and B.E. degree from Harbin Institute of Technology in 2009. Her research interests cover a wide range of areas including Internet of Things, artificial intelligence, mobile edge computing, and 5G.

Jiajia Liu (S'11-M'12-SM'15) is currently a full professor at the School of Cyber Engineering, Xidian University. His research interests cover wireless mobile communications, FiWi, IoT, etc. He has published more than 50 peer-reviewed papers in many prestigious IEEE journals and conferences, and currently serves as the associate editors for IEEE TC & TVT, the editor for IEEE Network, the guest editors of IEEE TETC & IEEE IoT Journal. He is a Distinguished Lecturer of IEEE ComSoc.



Yanlin Yue (SM'17) received his B.Eng degree in geographic information science and technology from China University of Geosciences, Wuhan, China, in 2015. He is working toward the M.Sc. degree at the Xidian University, China. His research interests include network economics for mobile edge computing, the optimization of wireless communication, and machine learning.

Haibin Zhang (M'16) received his Ph.D. degrees in computer science and technology from Xidian University in 2007, and B.S. degree in applied mathematics from Ocean University of China in 2003. He is currently an associate professor with the School of Cyber Engineering of Xidian university. His research interests concentrate on formal verification, wireless sensor networks, Internet of things.