# Truthful Online Double Auctions for Mobile Crowdsourcing: An On-demand Service Strategy

Shumei Liu, Yao Yu, *Member, IEEE,* Lei Guo, *Senior Member, IEEE,* Phee Lep Yeoh, *Member, IEEE,* Qiang Ni, *Senior Member, IEEE,* Branka Vucetic, *Life Fellow, IEEE,* and Yonghui Li, *Fellow, IEEE*

***Abstract*—Double auctions play a pivotal role in stimulating active participation of a large number of users comprising both task requesters and workers in mobile crowdsourcing. However, most existing studies have concentrated on designing offline two-sided auction mechanisms and supporting single-type tasks and fixed auction service models. Such works ignore the need of dynamic services and are unsuitable for large-scale crowdsourcing markets with extremely diverse demands (i.e., types and urgency degrees of tasks required by different requesters) and supplies (i.e., task skills and online durations of different workers). In this paper, we consider a practical crowdsourcing application with an on-demand service strategy. Especially, we innovatively design three online service models, namely online single-bid single-task (OSS), online single-bid multiple-task (OSM) and online multiple-bid multiple-task (OMM) models to accommodate diversified tasks and bidding demands for different users. Furthermore, to effectively allocate tasks and facilitate bidding, we propose a truthful online double auction mechanism for each service model based on the McAfee double auction. By doing so, each user can flexibly select auction service models and corresponding auction mechanisms according to their current interested tasks and online duration. To illustrate this, we present a three-demand example to explain the effectiveness of our on-demand service strategy in realistic crowdsourcing applications. Moreover, we theoretically prove that our mechanisms satisfy truthfulness, individual rationality, budget balance and consumer sovereignty. Through extensive simulations, we show that our mechanisms can accommodate the various demands of different users and improve social utility including platform utility and average user utility.**

***Index Terms*—Mobile crowdsourcing, online double auction, truthful mechanism design, on-demand service.**

## I. Introduction

CROWDSOURCING is an efficient problem-solving paradigm, which integrates public efforts to solve large-scale complex tasks that are challenging for an individual or business [1]. Due to the emergence of the Internet of Things (IoT) and the popularity of wireless smart devices, mobile crowdsourcing is attracting significant interests from research and industry to develop innovative applications. For example, mobile crowdsourcing can be harnessed to achieve large-scale wireless network coverage with low cost [2], provide high-quality training data for artificial intelligence [3] and build excellent location-based services for mobile users [4], [5].

Mobile crowdsourcing consists of two main roles, namely crowdsourcing platform and users including task requesters and workers. The platform provides an interface for task requesters to outsource complex tasks that need to be completed by workers with specific task skills [6]. One critical issue in practical mobile crowdsourcing is how to stimulate users to participate in the applications [7], [8], and auction-based incentive mechanisms are proven successful approaches to address this [9]. In a real-world crowdsourcing application, both crowdsourcing platform and users are self-interested and want to maximize their own benefits strategically. An untruthful auction is vulnerable to price manipulation and users may choose not to participate in fear of unfair treatment [10]. As such, a truthful auction-based incentive mechanism will encourage fair decisions and promote user participation in crowdsourcing [11].

More importantly, most auction-based incentive mechanisms for mobile crowdsourcing mainly aim to recruit enough workers and ignore the need to attract a sufficient number of task requesters [12], [13]. It is typically assumed that the task requesters will voluntarily issue their tasks to crowdsourcing platforms [14]. In [12]-[14], the authors default to a large amount of requesters' involvement, and then present a game-theoretic mechanism to incentivize the competitive and selfish workers to provide high-quality solutions. In reality, requesters may be unwilling to participate in crowdsourcing platforms due to various reasons such as unsatisfactory service experiences or communication security and privacy concerns [15]-[17]. Clearly, offering sufficient tasks is a fundamental requirement for recruiting more workers and determining the sustainability of a crowdsourcing application [14]. This motivates the need to further develop a two-sided incentive market that attracts both task requesters and workers for mobile crowdsourcing, which is more practical and fundamentally different from the traditional one-sided incentive mechanisms.

Some recent works have proposed two-sided incentive mechanisms for mobile crowdsourcing [18]-[24]. Amongst them, double auctions [19]-[21] are widely considered in a two-sided market with multiple buyers (task requesters) and

Manuscript received July 12, 2021; revised November 11, 2021 and December 16, 2021; accepted February 7, 2022.

S. Liu and Y. Yu are with both the School of Computer Science and Engineering, Northeastern University, and Key Laboratory of Intelligent Computing in Medical Image, Ministry of Education, Northeastern University, Shenyang 110819, China (E-mail: liusmneu@163.com; yuyao@mail.neu.edu.cn).

L. Guo is with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (E-mail: guolei@cqupt.edu.cn).

P. L. Yeoh, B. Vucetic and Y. Li are with the School of Electrical and Information Engineering, The University of Sydney, Sydney NSW 2006, Australia (E-mail:{phee.yeoh; branka.vucetic; yonghui.li}@sydney.edu.au).

Q. Ni is with the School of Computing and Communications and Data Science Institute, Lancaster University, UK (E-mail: q.ni@lancaster.ac.uk).

sellers (workers), where the McAfee double auction [18] is a proven successful paradigm for balancing market demands due to its essentially truthful property. Most existing works on double auctions in mobile crowdsourcing focus on offline scenarios, and only a few works consider online scenarios. Specifically, offline two-sided incentive mechanisms assume that the interest tasks and service strategies of users are fully known and do not change over time [25]. For instance, the authors in [19]-[23] consider an offline crowdsourcing scenario where the task requesters and workers arrive at the system in advance, and then study two-sided incentive mechanisms for crowdsourcing applications. We note that the offline solutions are more suitable to non-mobile crowdsourcing applications, where the task demands and number of workers do not change significantly over time [26]. However, users in realistic mobile crowdsourcing applications should arrive online in a random order, and their interests and status may frequently change over time [27].

To this end, only a few online two-sided solutions have been proposed for crowdsourcing applications. For each user arrival in practical online mechanisms, the platform must make an irrevocable decision about whether to select the user without knowledge of future user information. In [24], an online two-sided task allocation problem was designed for spatial crowdsourcing, but the authors did not consider the diversity of users' requirements in realistic crowdsourcing markets. More importantly, this solution ignored the selfishness of users, and thus did not satisfy the time and cost truthfulness. In [18], a truthful online double auction mechanism was proposed for dynamic mobile crowdsourcing. The authors in [18] assumed that the requested tasks of the same requester are identical, and a task is allocated only to a single worker, and completed in a time solt. The above two-sided solutions assume that the system has only a single type of task in a fixed auction service model and do not consider the various task requirements of different users. In other words, this auction mechanism can only serve users with fixed tasks and task skills. Moreover, some complex tasks may require multiple workers to perform or request many task results for further task analysis. As such, we highlight that there are significant gaps in applying existing solutions to practical mobile crowdsourcing scenarios.

We consider that an effective and practical crowdsourcing application must support multiple types of tasks simultaneously. Besides, in realistic mobile crowdsourcing markets, the demands and supplies are extremely diverse, where demands refer to the type and number of tasks required by different requesters and supplies are the task skills and online durations of different workers. Here, the demands of requesters and the supplies of workers can be regarded as the users' interests. As multiple requesters (or workers) with different interests may compete for potential capacities (or tasks), it is unsuitable to regard all interests as a whole and apply a single auction model for various user interests in a monopolistic way. This is because the users' interests are diverse, and each user's bidding requirement is unique. Therefore, a monopolistic approach cannot effectively match the demands and supplies of crowdsourcing markets, and thus reduces the efficiency of task allocation and the probability of successful bidding. For example, a requester with different types of tasks and urgency degree, and a worker with multiple task skills and relatively long online duration will prefer to choose a heterogeneous auction model to characterize and satisfy their interests in more detail. Unfortunately, the fixed auction model considered in existing online two-sided solutions [18]-[24] cannot support such on-demand services. As such, no existing online double auction mechanisms can satisfy the multiple-interest and on-demand auction situations.

In this paper, motivated by limitations of the state-of-art online double auction mechanisms for mobile crowdsourcing, we consider a practical case where multiple requesters demand different workforce and multiple workers supply various capacities in each time slot. Both requesters and workers participate in the crowdsourcing applications in a random pattern, and their interests and status change over time. We aim to provide an on-demand service strategy for different crowdsourcing users. Specifically, we design three online service models to accommodate the diversity of crowdsourcing demands and the variability of users' interests. Furthermore, we propose a truthful online incentive mechanism based on the McAfee double auction [28] for each service model, in which task pricing and winner selection are determined by market supply and demand. To the best of our knowledge, we are the first to consider a crowdsourcing application with multi-type tasks and design an on-demand online double auction strategy for mobile crowdsourcing according to the various interests of users. The joint consideration of various demands among users and spatio-temporal heterogeneity in the crowdsourcing market are significant challenges we address in our design. The main contributions of this paper are as follows:

- We design three requirement-based online auction service models, namely online single-bid single-task (OSS) model, online single-bid multiple-task (OSM) model and online multiple-bid multiple-task (OMM) model to meet various crowdsourcing bidding and task demands including different types and urgency degrees.
- Based on the McAfee double auction, we propose three truthful online double auction mechanisms for the above service models, which can effectively facilitate bidding and price tasks according to the demands and supplies of crowdsourcing markets.
- We present an application example to explain the practicability of our on-demand service strategy. We specifically consider three demand scenarios in Internet of Vehicles (IoV) environments, namely, single-task demand, single-type multiple-task demand and multiple-type multiple-task demand. The example clearly highlights the effectiveness of our solutions.
- We prove that our proposed incentive mechanisms satisfy time-truthfulness, cost-truthfulness, individual-rationality, budget-balance and consumer sovereignty. We further show that our solutions can support various demands of different users and improve social utility including platform utility and average user utility.

The rest of this paper is organized as follows. In Section II, we introduce the online incentive model and economic properties. In Section III, we design three online double auction models. Based on these models, we propose three truthful online double auction mechanisms in Section IV and present specific algorithms for each mechanism. In Section V, we introduce an application example to explain our practicability in real world. In Section VI, we provide simulations of the proposed mechanisms highlighting their advantages. The conclusion is given in Section VII.

## II. Online Double Auction Model and Economic Properties

In this section, we introduce our McAfee-based online double auction model for mobile crowdsourcing, and desirable properties of truthful online auction mechanisms.

### A. Online Double Auction Model for Mobile Crowdsourcing

In a crowdsourcing application, participating users include task requesters and workers, and any crowdsourcing applications require not only adequate tasks but also sufficient workers to complete the tasks. We therefore focus on the two-sided incentive of task requesters and workers by using McAfee double auction. McAfee double auction can accommodate market demand and ensure the truthfulness of auction markets. Specifically, we model the interactive procedure between task requesters, workers and crowdsourcing platform as a double auction, where task requesters are buyers and workers are sellers, and the platform is the auctioneer. Fig. 1 shows our McAfee-based online double auction model for mobile crowdsourcing with the dynamic arrival of users over time.

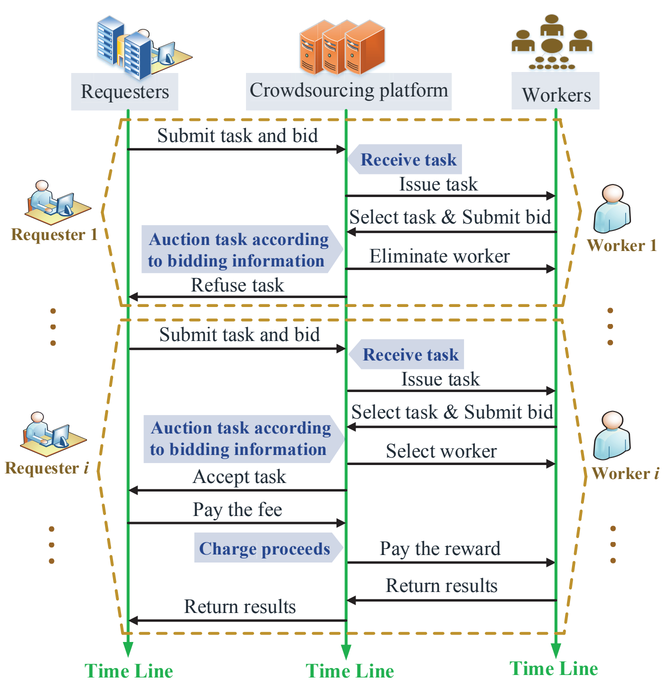



Fig. 1: Online double auction model for mobile crowdsourcing

In Fig. 1, the interactive process between the platform and the participating users (i.e., task requesters and workers) follows an online auction. Specifically, the McAfee double auction [29] is used to motivate task requesters and workers in an online fashion. In this case, both requesters and workers are randomly joining the platform anytime. Once a requester or a worker arrives, the platform has to make irrevocable decisions based on the current situation on whether to select the user as a winner and how much the user should pay or receive. The critical parts of McAfee double auction are as follows.

(1) Bids sorting: Sort the bids $B^b$ of buyers in non-increasing order, and sort the bids $B^s$ of sellers in non-decreasing order:

$$B_1^b \geq B_2^b \geq ... \geq B_h^b \geq ... \geq B_M^b, \tag{1}$$

$$B_1^s \leq B_2^s \leq ... \leq B_h^s \leq ... \leq B_N^s. \tag{2}$$

(2) Winners selection: Find the index of the least profitable transaction, $h = \arg\max \left(B_h^b \geq B_h^s\right)$. The first $h-1$ buyers and sellers are the auction winners. Specifically, in the sorted lists of buyers and sellers, we need to ensure that the bid of the $i$th winning buyer is higher than that of the $i$th winning seller. Here, $h$ is the maximum value of $i$. As such, the first $h-1$ buyers in (1) will be the winning buyers, and the first $h-1$ sellers in (2) will be the winning sellers.

(3) Pricing: Charge all the winning buyers equally according to the bid of the $h$th buyer $B_h^b$. Pay all winning sellers equally with the bid of the $h$th seller $B_h^s$.

For task requesters, each requester can submit tasks and tasks' bids (i.e., fees for obtaining the results of tasks) on the platform at any time. When a task requester arrives, the crowdsourcing platform must immediately decide whether to accept the requester's tasks at this time, and if so, which workers will be assigned to perform the tasks. For workers, each worker expects a payment in return for completing tasks. When a worker arrives, he/she can select the tasks he/she is good at and submits the corresponding bids (i.e., rewards for completing the tasks) to the platform. Then, the crowdsourcing platform must immediately decide whether to select the worker according to the current worker information, and if so, at what price.

For both the task requesters and the workers, the winners of each auction are selected based on comparing users' bids with the dynamically updated bidding thresholds. Specifically, if a requester's unit bid for his/her submitted task is higher than the bidding threshold set for task requesters, the task is selected. Similarly, if a worker's bid for his/her interested task is lower than the bidding threshold set for workers, the worker is selected to complete the task. The setting and updating of the bidding thresholds are particularly important to ensure the truthfulness of the online incentive mechanism for crowdsourcing. It is necessary to analyze bids of all existing users to obtain reasonable thresholds for task requesters and workers.

To determine the bidding thresholds according to the market changes and demands, we consider a multi-stage sampling-accepting and threshold-updating model, which is widely used in existing works on online auction [30], and shown in Fig. 2.

As commonly adopted in previous online auction solutions for crowdsourcing [18], [30], to better describe the user

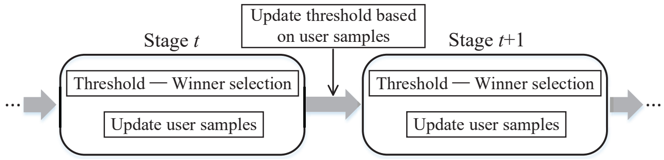



Fig. 2: Multiple-stage sampling-accepting and threshold-updating process

requests, our considered online incentive model is assumed to operate in a slotted structure and its timeline is discretized into time slots $T = \{1, 2, 3, ...., t, ...\}$. At each discrete time slot, each participating crowdsourcing user sends an interested task bidding request to the platform, then the platform will gather all bids information and determine the winners for each involved task based on the current global system information. The multiple-stage model dynamically updates the user samples including requester samples and worker samples, and then updates the bidding thresholds used for future decisions. In our online model, "slot" is equivalent to "stage". In each slot, the platform firstly selects winners including task requesters and workers according to the current bidding thresholds. Then, the platform updates the user samples by adding all winners of the slot to the existing user sample. After that, the bidding thresholds are updated according to the bids information of current winners, and used to select winners in the next slot. We use McAfee double auction to set and update the thresholds, and the specific technical details are provided in Section IV.

### B. Objective of Economic Properties

The objective of our truthful online double auction mechanisms is to satisfy time truthfulness, cost truthfulness, individual rationality, budget balance and consumer sovereignty. The five economic properties are critical to design economic-robust and market-truthful online mechanism. We now define the properties.

(1) Time truthfulness: An online auction mechanism is time-truthful if any user's utility is maximized when providing his/her real arrival and departure time.

(2) Cost truthfulness: An online auction mechanism is cost-truthful if any user's utility is maximized when their bids are equal to the true values or costs of tasks. In other words, each requester's bid is equal to the true value of his/her task, and each worker's bid is equal to the true cost for completing the task. In this case, no user can improve his/her utility by submitting a false bid.

(3) Individual rationality: An online auction mechanism is individually rational if each user has a non-negative utility when bidding its true bid. That is, each requester's payment is not more than his/her bid, and each worker's reward is not less than his/her bid.

(4) Budget balance: An online auction mechanism is budget-balance if the utility of the platform is non-negative.

(5) Consumer sovereignty: An online auction mechanism guarantees consumer sovereignty if the mechanism cannot arbitrarily exclude a user. Hence, a requester's task must be accepted by the platform if his/her bid is high enough. Also, a worker must be selected by the platform and obtain a payment if his/her bid is low enough.

TABLE I: List of Key Variables

| Notations | Definitions |
|---|---|
| $B_m^r$,$B_n^w$ | Bidding information of requester $r_m$ and worker $w_n$ |
| $b_m^r$,$v_m^r$ | Bid and true value of $r_m$ for the submitted task(s) |
| $b_n^w$,$c_n^w$ | Bid and true cost of $w_n$ for the interested task(s) |
| $at_m^r$,$dt_m^r$ | Submitted arrival and departure time of requester $r_m$ |
| $at_w^n$,$dt_w^n$ | Submitted arrival and departure time of $w_n$ |
| $\overline{a}t$,$\overline{b}t$ | True arrival and departure time |
| $\mu$ | Urgency degree of submitted task(s) by requesters |
| $R_t$,$W_t$ | Sets of participating requesters and workers in slot $t$ |
| $R_{W,t}$,$W_{W,t}$ | Sets of wining requesters and winning workers |
| $q_m$,$p_n$ | Pricing of requester $r_m$ and worker $w_n$ |
| $u_{r,m}$,$u_{w,n}$ | Utilities of requester $r_m$ and worker $w_n$ |
| $u_p$ | Utility of the crowdsourcing platform |
| $U_t^k$ | Budget of task $\tau_k$ for recruiting workers in slot $t$ |
| $\delta_R^k$ | Bidding threshold for requesters to task $\tau_k$ |
| $\delta_W^k$ | Bidding threshold for workers to task $\tau_k$ |
| $\lambda_R^k$ | Task demand threshold of task $\tau_k$ |
| $\lambda_W^k$ | Worker demand threshold of task $\tau_k$ |

Among these properties, truthfulness is the most crucial property in auction theory [11].

## III. Requirement-Based Online Auction Service Models

In this section, we design three requirement-based online crowdsourcing models to meet the different bidding demands of users, which are the basis for online double auction mechanisms discussed in the next section. Our first model is the OSS model which allows each user to submit a bid containing a task. The OSS model is a common model in existing crowdsourcing services. Our second model is the OSM model which allows each user to submit a bid containing multiple tasks. It can achieve the multi-task requirements of users in a biding process, and improve the service efficiency of crowdsourcing. Our third model is the OMM model. It allows each user to submit multiple bids at a time, with each bid containing multiple tasks.

We assume that there is a set of task requesters $R = \{r_1, r_2, ..., r_m, ..., r_M\}$ and a set of workers $W = \{w_1, w_2, ..., w_n, ...w_N\}$ in each slot, where $M$ and $N$ are the number of requesters and workers, respectively. Each requester $r_m$ in each slot has a set of tasks $J_m = \left\{j_m^1, j_m^2, ...\right\}$ that need to be completed, and each worker $w_n$ in each slot has a set of interested tasks $\Gamma_n = \left\{t_n^1, t_n^2, ...\right\}$. Next we will introduce our three online crowdsourcing models in details. For ease of reference, we list important notations in Table 1.

### A. The OSS Model: Online Single-bid Single-task

First, we introduce the relevant application scenarios of the OSS model according to the users' demands. For requesters, they have only one current request task. For workers, they only have a single task skill, and their online duration is short.

To quickly achieve task allocation and auction bidding, these users adopt the OSS model to submit their interested tasks and corresponding bids. In the OSS model, each requester $r_m$ or worker $w_n$ can only submit one bid to the platform, and task set $J_m = \{j_m\}$ or $\Gamma_n = \{t_n\}$ in the bid contains only one task. The bidding information $B_m^r$ and $B_n^w$ of each requester $r_m$ and each worker $w_n$ in the OSS model are shown in Fig. 3.

$$B_m^r \boxed{at_m^r, dt_m^r, J_m[j_m(\lambda,\mu)], b_m^r} \qquad B_n^w \boxed{at_n^w, dt_n^w, \Gamma_n[t_n], b_n^w}$$

The bid of requester $r_m$ The bid of worker $w_n$

Fig. 3: The bidding information in the OSS model

Based on the bidding information of the OSS model, we next detail the workflow and the utility of each crowdsourcing entity in the OSS model.

- Crowdsourcing workflow in the OSS model

In the OSS crowdsourcing model, users arrive and depart dynamically. When task requester $r_m$ arrives, he/she submits the bidding information $B_m^r$ to the platform, as shown in Fig. 3. In the bidding information $B_m^r = (at_m^r, dt_m^r, J_m[j_m(\lambda)], b_m^r)$ of requester $r_m$, $at_m^r \in T$ and $dt_m^r \in T$ represent the submitted arrival and departure time of task requester $r_m$, respectively. The online duration of requester $r_m$ is from $at_m^r$ to $dt_m^r$. $J_m$ needs to be completed before the departure time of $r_m$, and $\lambda$ is the specific demand of task $j_m$, such as the number of required results (i.e., the specific number of workers required for the task). $\mu$ is the urgency degree of task $j_m$, and can be represented by the average tolerance delay of the task. $b_m^r$ is the bid of requester $r_m$ for task $j_m$. The true value of the submitted task by requester $r_m$ is denoted by $v_m^r$ and is considered to be private information of requester $r_m$. The relationship between $b_m^r$ and $v_m^r$ is $b_m^r \leq v_m^r$. In bidding information $B_n^w = (at_n^w, dt_n^w, \Gamma_n[t_n], b_n^w)$ of worker $w_n$, $at_n^w \in T$ and $dt_n^w \in T$ represent the submitted arrival and departure time of worker $w_n$, respectively. The active duration of worker $w_n$ is from $at_m^r$ to $dt_m^r$. $t_n$ is the interested task of worker $w_n$ and $b_n^w$ is the bid of worker $w_n$ for task $t_n$. $c_n^w$ is the true cost for completing the task submitted by worker $w_n$ and is considered to be private information of worker $w_n$. The relationship between $b_n^w$ and $c_n^w$ is $b_n^w \geq c_n^w$.

The platform determines a bidding threshold $\delta_R$ for task requesters, and $r_m$ with a bid $b_m^r \geq \delta_R$ is selected as a winner. For each discrete time slot $t \in T$, the platform selects a set of winners $R_W \subseteq R$. Then, the platform decides the fee $q_m \leq b_m^r$ for each requester $r_m \in R_W$ and returns the task's result before they depart. Similarly, the platform learns a bidding threshold $\delta_W$ for workers, and worker $w_n$ with a bid $b_n^w \leq \delta_W$ is selected as a winner. For each discrete time $t \in T$, the platform selects a set of winner $W_W \subseteq W$. Then, the platform decides reward $p_n \geq b_n^w$ for each worker $w_n \in W_W$ and receives the task's result before they depart.

- Crowdsourcing utility in the OSS model

We calculate the utility of each participating crowdsourcing entity in the OSS model. To facilitate the discussion, we introduce the following definitions.

$y_m$ is the request indicator of requester $r_m$ for bid $B_m^r$. If requester $r_m$'s task $J_m$ is accepted by the platform, $y_m = 1$, otherwise, $y_m = 0$. $x_n$ is the completion indicator of worker $w_n$ for bid $B_n^w$. If task $\Gamma_n$ is completed by worker $w_n$, $x_n = 1$, otherwise, $x_n = 0$. $q_m$ is the pricing for requester $r_m \in R_W$, i.e., the fee that the platform collects from requester $r_m$, and if $r_m \notin R_W$, $q_m = 0$. $p_n$ is the pricing for worker $w_n \in W_W$, i.e., the reward that the platform pays worker $w_n$, and if $w_n \notin W_W$, $p_n = 0$.

Based on the above parameter definitions, in each time slot, the utility $u_{r,m}^{\text{SS}}$ of requester $r_m$ in the OSS model is

$$u_{r,m}^{\text{SS}} = y_m v_m^r - q_m. \tag{3}$$

The utility $u_{w,n}^{\text{SS}}$ of worker $w_n$ in each time slot is

$$u_{w,n}^{\text{SS}} = p_n - x_n c_n^w. \tag{4}$$

The utility $u_p^{\text{SS}}$ of the platform in each time slot is

$$u_p^{\text{SS}} = \sum\nolimits_{m=1}^{M} q_m - \sum\nolimits_{n=1}^{N} p_n. \tag{5}$$

### *B. The OSM Model: Online Single-bid Multiple-task*

First, we detail the relevant application scenarios of the OSM model according to the users' demands. For requesters, they have multiple current tasks with the same type and urgency degree. For workers, they only have a single task skill, and their online duration is relatively long. To quickly achieve task allocation and auction bidding, these users choose the OSM model to submit their interested tasks and corresponding bids. In the OSM model, each requester $r_m$ or worker $w_n$ can only submit one bid to the platform, and task sets $J_m = \{j_m^1, j_m^2, ...\}$ or $\Gamma_n = \{t_n^1, t_n^2, ...\}$ in the bidding information contain multiple tasks. In Fig. 4, we show the bidding information $B_m^r$ and $B_n^w$ of $r_m$ and $w_n$ in the OSM model, assuming that the submitted bid contains four tasks. $\mu$ is the urgency degree of these tasks.

$$B_m^r \boxed{at_m^r, dt_m^r, J_m[j_m^1(\lambda_1), j_m^2(\lambda_2), j_m^3(\lambda_3), j_m^4(\lambda_4), \mu], b_m^r}$$

The bid of requester $r_m$

$$B_n^w \boxed{at_n^w, dt_n^w, \Gamma_n[t_n^1, t_n^2, t_n^3, t_n^4], b_n^w}$$

The bid of worker $w_n$

Fig. 4: The bidding information in the OSM model

Unlike the OSS model, the OSM model can achieve the multi-task requirements of each user in a biding process, which improves the efficiency of crowdsourcing. Since both the OSS and OSM models can submit only one bid, the workflow of crowdsourcing and the utility of each entity in crowdsourcing in the OSM model are the same as those in the OSS model defined in the previous subsection.

### *C. The OMM Model: Online Multiple-bid Multiple-task*

First, we discuss the relevant application scenarios of the OMM model according to the users' demands. For requesters, they currently have multiple types of tasks, with the same urgency degrees for each type of tasks. For workers, they have multiple task skills, and their online duration is relatively long. To characterize their interests in more detail, these users choose the OMM model to submit their interested tasks and corresponding bids. Therefore, rather than submitting a single bid as in the OSS and OSM models, in the OMM model, each requester $r_m$ or worker $w_n$ can submit multiple bids to the platform at a time, and each bid can contain one or multiple tasks. In Fig. 5, we show the bidding information $B_m^r$ and $B_n^w$ of $r_m$ and $w_n$ in the OMM model, and we assume that there are three bids per time.

| | | | |
|---|---|---|---|
| $B_m^r(1)$ | $at_m^r, dt_m^r, J_m^{(1)}[j_m^1(\lambda_1), j_m^2(\lambda_2), \mu_1], b_m^r(1)$ | $B_n^w(1)$ | $at_n^w, dt_n^w, \Gamma_n^{(1)}[t_n^1, t_n^2], b_n^w(1)$ |
| $B_m^r(2)$ | $at_m^r, dt_m^r, J_m^{(2)}[j_m^3(\lambda_3), j_m^4(\lambda_4), \mu_2], b_m^r(2)$ | $B_n^w(2)$ | $at_n^w, dt_n^w, \Gamma_n^{(2)}[t_n^2, t_n^3], b_n^w(2)$ |
| $B_m^r(3)$ | $at_m^r, dt_m^r, J_m^{(3)}[j_m^5(\lambda_5), j_m^6(\lambda_6), \mu_3], b_m^r(3)$ | $B_n^w(3)$ | $at_n^w, dt_n^w, \Gamma_n^{(3)}[t_n^1, t_n^2, t_n^3], b_n^w(3)$ |
| | The bid of requester $r_m$ | | The bid of worker $w_n$ |

Fig. 5: The bidding information in the OMM model

- Crowdsourcing workflow in the OMM model

Based on the bidding information of the OMM model, we next detail the workflow and the utility of each crowdsourcing entity in the OMM model.

In the OMM crowdsourcing model, users arrive and depart platform dynamically. When task requester $r_m$ arrives, he/she submits multiple (i.e., three in our example) bids $B_m^r = \{B_m^r(1), B_m^r(2), B_m^r(3)\}$ to the platform, where $B_m^r(i) = (at_m^r, dt_m^r, J_m^{(i)}, b_m^r)$ represents $i$th bid submitted by $r_m$. $J_m^{(i)}$ is the corresponding task set that contains multiple tasks, $b_m^r(i)$ is the bid of requester $r_m$ for task $J_m^{(i)}$, and $v_m^r(i)$ is the true value of $J_m^{(i)}$. The relationship between $b_m^r(i)$ and $v_m^r(i)$ is $b_m^r(i) \leq v_m^r(i)$. $\mu_i$ is the urgency degree of different types of tasks. Similarly, when worker $w_n$ arrives, he/she submits multiple (i.e., three in our example) bids $B_n^w = \{B_n^w(1), B_n^w(2), B_n^w(3)\}$ to the platform, where $B_n^w(i) = (at_n^w, dt_n^w, \Gamma_n^{(i)}, b_n^w(i))$ represents $i$th bid submitted by worker $w_n$. $\Gamma_n^{(i)}$ is the corresponding task set that contains multiple tasks, and $b_n^w(i)$ is the bid of worker $w_n$ for task $\Gamma_n^{(i)}$. $c_n^w(i)$ is the true cost of $\Gamma_n^{(i)}$. The relationship between $b_n^w(i)$ and $c_n^w(i)$ is $b_n^w(i) \geq c_n^w(i)$.

Note that the winner selection and pricing (i.e., determining fee for requesters and reward for workers) for each bid in the OMM model is the same as the OSS model.

- Crowdsourcing utility in the OMM model

Next, we calculate the utility of each crowdspurcing entity in the OMM model. To facilitate the discussion, we introduce the following definitions where $I_m^r$ is the number of bids submitted by requester $r_m$ at each time and $I_n^w$ is the number of bids submitted by worker $w_n$ at each time:

$y_m^{(i)}$ is the request indicator of requester $r_m$ for bid $B_m^r(i)$. If requester $r_m$'s task $J_m^{(i)}$ in bid $B_m^r(i)$ is accepted by the platform, $y_m^{(i)} = 1$, otherwise, $y_m^{(i)} = 0$. $x_n^{(i)}$ is the completion indicator of worker $w_n$ for bid $B_n^w(i)$. If task $\Gamma_n^{(i)}$ is completed by worker $w_n$, $x_n^{(i)} = 1$, otherwise, $x_n^{(i)} = 0$. $q_m^{(i)}$ is the fee that the platform collects from $r_m \in R_W$ for bid $B_m^r(i)$. If bid $B_m^r(i)$ is not accepted by the platform, $q_m^{(i)} = 0$. $q_m$ is the total fees that the platform collects from $r_m \in R_W$ for total bids $B_m^r$ at a time, i.e., $q_m = \sum_{i=1}^{I_m^r} q_m^{(i)}$. $p_n^{(i)}$ is the reward that the platform pays worker $w_n \in W_W$ for bid $B_n^w(i)$. If bid $B_i^n$ is not selected by the platform, $p_n^{(i)} = 0$. $p_n$ is the total fees that the platform pays worker $w_i \in W_W$ for the total bids $B_n^w$ at a time, i.e., $p_n = \sum_{i=1}^{I_n^w} p_n^{(i)}$.

Based on the above parameter definitions, in each time slot, the utility $u_{r,m}^{\text{MM}}$ of requester $r_m$ in the OMM model is

$$u_{r,m}^{\text{MM}} = \sum\nolimits_{i=1}^{I_m^r} y_m^{(i)} v_m^r(i) - q_m. \tag{6}$$

The utility $u_{w,n}^{\text{MM}}$ of worker $w_n$ in each time slot is

$$u_{w,n}^{\text{MM}} = p_n - \sum\nolimits_{i=1}^{I_n^w} x_n^{(i)} c_n^w(i). \tag{7}$$

The utility $u_p^{\text{MM}}$ of the platform in each time slot is

$$u_p^{\text{MM}} = \sum\nolimits_{m=1}^{M} q_m - \sum\nolimits_{n=1}^{N} p_n. \tag{8}$$

Based on the above, each task requester or worker can flexibly select auction service models according to the current interested type of tasks, number of tasks, urgency degrees of tasks, and online durations. As such, a requester with an urgent task or a worker with a single task skill and short online duration will choose the OSS model. A requester with multiple tasks of the same (different) type(s) and urgency degree or a worker with a single (multiple) task skill(s) and relatively long online time will choose the OSM (OMM) model.

## IV. The Proposed Truthful Online Double Auction Mechanisms

In this section, based on McAfee double auction, we propose three truthful online double auction mechanisms (TODAs) for the above three auction service models, namely TODA-SS, TODA-SM and TODA-MM mechanisms.

As described above, one of the most critical features of our truthful online double auction mechanisms is to set reasonable thresholds for task requesters and workers in each time slot. For task requesters, the thresholds we need to set are the bidding threshold for requesters and task demand threshold (i.e., task requesters' demand for each submitted task). For workers, the thresholds we need to set are the bidding threshold for workers and worker demand threshold (i.e, the number of workers required for completing each task). McAfee double auction is an excellent mechanism to set and dynamically update these thresholds over time according to the market supply and demand.

### A. TODA-SS Auction Mechanism

When a user currently has only one interested task and the online duration is short, he/she adopts the OSS model and TODA-SS auction mechanism to quickly achieve task allocation and auction bidding. We firstly introduce the TODA-SS auction mechanism. The TODA-SS mechanism consists of three aspects: winner selection, task pricing and threshold updating. In our online double auction mechanism, the time is discretized into $T = \{1, 2, 3, ...., t, ...\}$ slots. At each slot

$t$, we perform winner selection. The platform quickly selects qualified task requesters and workers (i.e., winners) based on the thresholds mentioned above. More importantly, we then perform task pricing. The platform determines the fee for each winning requester and the reward for each winning worker. Then, we perform threshold updating. The platform dynamically updates the thresholds by using double auction to measure market changes and current users' demands. The updated thresholds are used to select qualified task requesters and workers (i.e., winners) in the next slot, i.e., slot $t+1$. In other words, the updated thresholds in slot $t$ are the prediction of the users' demands to each task in slot $t+1$.

For each slot $t$, $R_t$ and $W_t$ are defined as the currently participating task requesters and workers in this slot, respectively. Besides, for task $\tau_k$, $U_t^k$ is the budget of task $\tau_k$ for recruiting workers in slot $t$, which is the total amount of fees collected from task requesters for task $\tau_k$ in slot $t-1$. We define $\delta_R^k$ as the bidding threshold for task requesters to task $\tau_k$, which is the highest unit bid (i.e., $b_m^r/\lambda_m^k$) to task $\tau_k$ among all winning requesters in the previous slot $t-1$. $\delta_W^k$ is the bidding threshold for workers to task $\tau_k$, which is the highest bid (i.e., $b_n^w$) to task $\tau_k$ among all winning workers in the previous slot $t-1$. Also, $\lambda_R^k$ is the task demand threshold of task $\tau_k$, which is the maximum task demand (i.e., $\lambda_m^k$) to task $\tau_k$ among all winning requesters in the previous slot $t-1$. $\lambda_W^k$ is the worker demand threshold of task $\tau_k$, which is the number of winning workers required for completing task $\tau_k$ in the previous slot $t-1$. $Q_{R,t}=\{q_1,q_2,...\}$ is the set of fees for the winning requesters in set $R_{W,t}$ and $P_{W,t}=\{p_1,p_2,...\}$ is the set of rewards for the winning workers in set $W_{W,t}$.

- ***Winner Selection Algorithm in TODA-SS***

First, we introduce the winner selection process of our TODA-SS mechanism, which is shown in Algorithm 1. Steps 1-4 show the winner selection rules for requesters. $\lambda_m^k$ is the specific demands for task $\tau_k$ requested by requester $r_m$. To avoid wasting workers' resources due to excessive requests, we use $b_m^r/\lambda_m^k$ instead of just bid $b_m^r$ to determine whether requester $r_m$ is selected as a winner for task $\tau_k$. The winners among task requesters for task $\tau_k$ are those whose $b_m^r/\lambda_m^k$ is not less than $\delta_R^k$. In this case, a task requester is more likely to become a winner if he/she submits a higher bid and less demands for a task. Moreover, steps 5-10 show the winner selection rules for workers. The winners among workers for task $\tau_k$ are those whose $b_n^w$ is not more than bidding threshold $\delta_W^k$. Besides, bidding threshold $\delta_W^k$ is within budget constraint $U_t^k$ of task $\tau_k$ in this slot, and the number of workers required for completing $\tau_k$ is non-zero. Algorithm 1 outputs winner sets $R_{W,t}$ and $W_{W,t}$ of requesters and workers in slot $t$, respectively.

- ***Task Pricing Algorithm in TODA-SS***

Next, we introduce the task pricing process of our TODA-SS mechanism, which is shown in Algorithm 2. The pricing (i.e., fees) for winners among task requesters depends on bidding threshold $\delta_R^k$ and task demand threshold $\lambda_R^k$. If specific demand $\lambda_m^k$ for task $\tau_k$ by requester $r_m(r_m \in W_{W,t})$ is more than $\lambda_R^k$, the fee of $r_m$ is the product of $\lambda_R^k$ and $\delta_R^k$. Otherwise, the fee of $r_m$ is the product of $\lambda_m^k$ and $\delta_R^k$. The pricing (i.e., rewards) for each winner $w_n, (w_n \in R_{W,t})$ to task $\tau_k$ is $\delta_W^k$.

**Algorithm 1** *Winner Select-TODA-SS ($R_t$,$W_t$,$U_t^k$)*

**Input:** $R_t$, $W_t$, $U_t^k$, $\delta_R^k$, $\delta_W^k$, $\lambda_R^k$, $\lambda_W^k$
**Output:** winner sets $R_{W,t}$ and $W_{W,t}$
1: **for** each requester $r_m \in R_t$ **do**
2:     **for** task $\tau_k$ in $J_m$ **do**
3:         **if** $(b_m^r/\lambda_m^k) \geq \delta_R^k$ **then**
4:             $R_{W,t} \leftarrow R_{W,t} \cup r_m$
5: **for** each task $\tau_k$ **do**
6:     **while** $\lambda_W^k > 0$ **do**
7:         **for** all workers in set $W_t$ **do**
8:             $w_i \leftarrow \arg\min(b_n^w)$;
9:             **if** $b_n^w \leq \delta_W^k$ and $\delta_W^k \leq U_t^k$ **then**
10:                 $W_{W,t} \leftarrow W_{W,t} \cup w_i$, $W_t \leftarrow W_t \backslash w_i$, $U_t^k = U_t^k - \delta_W^k$, $\lambda_W^k = \lambda_W^k - 1$;

**Algorithm 2** *Pricing-TODA-SS ($R_{W,t}$,$W_{W,t}$)*

**Input:** $R_{W,t}, W_{W,t}$
**Output:** pricing sets $Q_{R,t}$ and $P_{W,t}$
1: **for** each $r_m \in R_{W,t}$ **do**
2:     **if** $\lambda_m^k > \lambda_R^k$ **then**
3:         $q_m^k = \lambda_R^k * \delta_R^k$;
4:     **else**
5:         $q_m^k = \lambda_m^k * \delta_R^k$;
6:     $q_m = q_m^k$;
7: **for** each $w_n \in W_{W,t}$ **do**
8:     $p_n^k = \delta_W^k$, $p_n = p_n^k$;

- ***Threshold Updating Algorithm in TODA-SS***

Then, we introduce the threshold updating process of our TODA-SS mechanism, which is shown in Algorithm 3. We aim to update four thresholds for each task $\tau_k$ according to the McAfee double auction, which are bidding threshold $\delta_R^k$ for requesters, bidding threshold $\delta_W^k$ for workers, task demand threshold $\lambda_R^k$ and worker demand threshold $\lambda_W^k$. Using the technical framework of McAfee double auction, for each task $\tau_k$, we obtain set $R_t^k$ by sorting $b_m^r/\lambda_m^k$ for the requesters in $R_t$, and set $W_t^k$ by sorting $b_n^w$ for the workers in $W_t$ as shown in steps 1-3. $R_t^k(h)$ and $W_t^k(h)$ represent $h$th element in sets $R_t^k$ and $W_t^k$, respectively. Then, we find $\max\left\{h|R_t^k(h) \geq W_t^k(h), h < \min\{M,N\}\right\}$, as shown in step 4. Finally, we update the thresholds for task $\tau_k$, as shown in steps 5-6. Specifically, at the end of slot $t$, the bidding threshold for task requesters is updated as the highest unit bid among all winning requesters in this slot. Also, the bidding threshold for workers is updated as the highest bid among all winning workers in this slot.

- ***Overall Algorithm of TODA-SS***

Finally, based on Algorithms 1, 2 and 3, we introduce the overall process of our proposed TODA-SS mechanism, which is shown in Algorithm 4.

We can find that at the beginning of slot $t=0$, the platform integrates tasks submitted by the all participating

**Algorithm 3** *Update Threshold-TODA-SS* ($R_t$,$W_t$)

**Input:** $R_t$,$W_t$

**Output:** updated thresholds $\delta_R, \delta_W, \lambda_R, \lambda_W$

1: **for** each task $\tau_k$ **do**
2: $\quad R_t^k \leftarrow$ sort $\{b_m^r/\lambda_m^k, r_m \in R_t, \tau_k \in J_m\}$ in non-increasing order;
3: $\quad W_t^k \leftarrow$ sort $\{b_n^w, w_m \in W_t, \tau_k \in \Gamma_n\}$ in non-decreasing order;
4: $\quad h_k \leftarrow \arg\max \left\{R_t^k(h) \geq W_t^k(h), h < \min\{M, N\}\right\}$;
5: $\quad \lambda_R^k \leftarrow \max\left\{\lambda_m^k, r_m \in \{R_t^k(1), R_t^k(2), ..., R_t^k(h_k - 1)\}\right\}$;
6: $\quad \lambda_W^k \leftarrow h_k - 1$, $\delta_R^k \leftarrow R_t^k(h_k)$, $\delta_W^k \leftarrow W_t^k(h_k)$;

**Algorithm 4** *TODA-SS* ($R_t, W_t, t$)

**Input:** $R_t, W_t, t$

**Output:** winner sets $R_{W,t}$ and $W_{W,t}$, pricing sets $Q_{R,t}$ and $P_{W,t}$, and updated thresholds $\delta_R, \delta_W, \lambda_R, \lambda_W$

1: Initialize: $(t, U_t, \delta_R, \delta_W, \lambda_R, \lambda_W) \leftarrow (0, 0, 0, \infty, 0, 0)$, $(P_W, Q_R) \leftarrow (\phi, \phi)$;
2: **while** $t = 0$ **do**
3: $\quad$ integrate participating requesters at slot $t = 0$ into $R_0$, $\mathcal{T}_0 \leftarrow \cup_{r_m \in R_0} J_m$;
4: $\quad$ integrate participating workers at slot $t = 0$ into $W_0$;
5: $\quad$ **for** each task $\tau_k \in \mathcal{T}_0$ **do**
6: $\quad\quad R_0^k \leftarrow$ sort $\{b_m^r/\lambda_m^k, r_m \in R_0, \tau_k \in J_m\}$ in non-increasing order;
7: $\quad\quad W_0^k \leftarrow$ sort $\{b_n^w, w_m \in W_0, \tau_k \in \Gamma_n\}$ in non-decreasing order;
8: $\quad\quad h_k \leftarrow \arg\max \left\{R_0^k(h) \geq W_0^k(h), h < \min\{M, N\}\right\}$;
9: $\quad\quad R_{W,0}^k \leftarrow w[R_0^k(i)], i = 1, 2, ..., h_k - 1$, $R_{W,0} \leftarrow R_{W,0} \cup R_{W,0}^k$;
10: $\quad\quad W_{W,0}^k \leftarrow r[W_0^k(i)], i = 1, 2, ..., h_k - 1$, $W_{W,0} \leftarrow W_{W,0} \cup W_{W,0}^k$;
11: $\quad\quad \lambda_R^k \leftarrow \max\left\{\lambda_m^k, r_m \in \{R_0^k(1), R_0^k(2), ..., R_0^k(h_k - 1)\}\right\}$, $\lambda_W^k \leftarrow h_k - 1$;
12: $\quad\quad \delta_R^k \leftarrow R_0^k(h_k)$, $\delta_W^k \leftarrow W_0^k(h_k)$;
13: $\quad\quad (P_{W,0}, Q_{R,0}) \leftarrow$ *Pricing-TODA-SS*($R_{W,0}$,$W_{W,0}$);
14: $\quad\quad U_0 = \sum_{r_m \in R_{W,0}} q_m - \sum_{w_n \in W_{W,0}} p_n$;
15: $\quad\quad$ integrate departing requesters at the end of slot $t = 0$ into $R_{d,0}$;
16: $\quad\quad$ integrate departing workers at the end of slot $t = 0$ into $W_{d,0}$;
17: $\quad\quad R_0 \leftarrow R_0 \backslash R_{d,0}$, $W_0 \leftarrow W_0 \backslash W_{d,0}$, $t = t + 1$;
18: **while** $t < T$ **do**
19: $\quad$ integrate participating requesters at slot $t$ into $R_t$, $\mathcal{T}_t \leftarrow \cup_{r_m \in R_t} J_m$;
20: $\quad$ integrate participating workers at slot $t$ into $W_t$;
21: $\quad$ **for** each task $\tau_k \in \mathcal{T}_t$ **do**
22: $\quad\quad (R_{W,t}, W_{W,t}) \leftarrow$ *Winner Select-TODA-SS*;
23: $\quad\quad (P_{W,t}, Q_{R,t}) \leftarrow$ *Pricing-TODA-SS*;
24: $\quad\quad (\delta_R, \delta_W, \lambda_R, \lambda_W) \leftarrow$ *Update Threshold-TODA-SS*;
25: $\quad\quad U_t = \sum_{r_m \in R_{W,t}} q_m$;
26: $\quad\quad$ integrate departing requesters at the end of slot $t$ into $R_{d,t}$;
27: $\quad\quad$ integrate departing workers at the end of slot $t$ into $W_{d,t}$;
28: $\quad\quad R_t \leftarrow R_t \backslash R_{d,t}$, $W_t \leftarrow W_t \backslash W_{d,t}$;
29: $\quad t = t + 1$;

requesters into a set $\mathcal{T}_0$. For each $\tau_k \in \mathcal{T}_0$, we apply McAfee double auction mechanism to select winners including winning requesters and workers, and obtain initial thresholds $\lambda_R^k$, $\lambda_W^k$, $\delta_R^k$ and $\delta_W^k$ as shown in steps 1-17. For each task $\tau_k \in \mathcal{T}_0$, we obtain set $R_0^k$ by sorting $b_m^r/\lambda_m^k$ for the requesters in $R_0$, and set $W_0^k$ by sorting the $b_n^w$ for the workers in $W_0$ as shown in steps 5-7. In steps 8-10, we select winners from participating requesters and workers. $r[R_0^k(i)]$ is the requester whose $b_m^r/\lambda_m^k$ is $R_0^k(i)$ and $w[W_0^k(i)]$ is the worker whose bid $b_n^w$ is $W_0^k(i)$. Then, we obtain initial thresholds $\lambda_R^k$, $\lambda_W^k$, $\delta_R^k$ and $\delta_W^k$ as shown in steps 11-12. Moreover, we calculate budget $U_0$ of the platform, which is used for recruiting workers in slot $t = 1$ as shown in steps 13-14. At the end of slot $t = 0$, the platform removes the departed requesters and workers from sets $R_0$ and $W_0$ as shown in steps 15-17. At the beginning of each slot $t > 0$, the platform integrates participating requesters and workers into $R_t$ and $W_t$, respectively. For each task $\tau_k \in \mathcal{T}_t$, we use *WinnerSelect-TODA-SS* (Algorithm 1) to select winners as shown in step 22. Next, we use *Pricing-TODA-SS* (Algorithm 2) to price the winners as shown in step 23. Then, we use *UpdateThreshold-TODA-SS* (Algorithm 3) to update four thresholds as shown in step 24. We calculate budget $U_t$ of the platform, which is used for recruiting workers at the next slot as shown in step 25. At the end of each slot $t > 0$, the platform removes the departed requesters and workers from sets $R_t$ and $W_t$ as shown in steps 26-28.

- ***Economic Properties Analysis of TODA-SS***

We now prove that our proposed TODA-SS auction mechanism satisfies the economic properties mentioned in Section II-B.

**Theorem 1.** *TODA-SS achieves time-truthfulness.*

*Proof.* For user $i$ (i.e., a participating requester or worker), we assume that his/her submitted arrival and departure time in the submitted bidding information are $at$ and $dt$, respectively. Besides, $\overline{at}$ and $\overline{dt}$ are his/her true arrival and departure time, respectively. In our proposed TODA-SS mechanism, the platform checks the active users and the departing users in each slot.

If user $i$ deliberately extends his/her actually active time on the platform, i.e., $\overline{at} < at < dt < \overline{dt}$, his/her utility will not increase when increasing his/her actually active time. This is because the platform checks departing users at the end of each slot according to their submitted bidding information instead of their actually active time. Even if user $i$ is still active before submitted arrival time $at$ and after submitted departure time $dt$, he/she will not be selected as the winner and his/her utility is 0 within two periods $(\overline{at}, at)$ and $(dt, \overline{dt})$. Therefore, users cannot improve their utility by extending the actually active time on the platform.

If user $i$ deliberately extends his/her submitted active time on the platform, i.e., $at < \overline{at} < \overline{dt} < dt$, requester $i$ will not be able to submit the tasks that needs to be performed or worker $i$ will not be able to complete the winning tasks within the period $(at, \overline{at})$. Also, requester/worker $i$ will cannot obtain/return the task results within the period $(\overline{dt}, dt)$. Then, his/her utility is 0 in these periods. Therefore, users cannot improve their utility by extending the submitted active time on the platform.

In conclusion, users cannot improve their utility by submitting dishonest active time. Therefore, the proposed TODA-SS auction mechanism satisfies time-truthfulness. □

**Theorem 2.** *TODA-SS achieves cost-truthfulness.*

*Proof.* We assume that all participating users cannot increase their utility by any means other than bidding. To prove TODA-SS's truthfulness, we need to prove that for any requester $r_m$ or worker $w_n$, he/she cannot improve his/her utility by bidding other than the true valuation of the task set. For this, we need to show that the winner selection is monotonic for both requesters and workers, and pricing is bid-independent.

**Monotonic winner selection**

The following two lemmas illustrate the monotonicity of TODA-SS's winner selection.

**Lemma 1:** Given $\{b_1^r, ..., b_m^r, ..., b_M^r\}$ for requesters and $\{b_n^w\}_{n=1}^N$ for workers, if requester $r_m$ wins the auction by bidding $b_m^r$, then he/she also wins by bidding $b_m^{r\prime} > b_m^r$.

**Lemma 2:** Given $\{b_1^w, ..., b_n^w, ...b_N^w\}$ for workers and $\{b_m^r\}_{m=1}^M$ from requesters, if worker $w_n$ wins the auction by bidding $b_n^w$, then he/she also wins by bidding $b_n^{w\prime} < b_n^w$.

*Proof.* Lemmas 1 and 2 can be proved by McAfee double auction, which is detailed in Section II-A. □

**Bid-independent pricing**

We show that pricing is bid-independent for both requesters and workers by the following two Lemmas.

**Lemma 3:** Given $\{b_1^r, ..., b_m^r, ..., b_M^r\}$ for requesters and $\{b_n^w\}_{n=1}^N$ for workers, if requester $r_m$ wins the auction by bidding $b_m^r$ or $b_m^{r\prime}$, then fee $q_m$ charged to $r_m$ is the same for both.

**Lemma 4:** Given $\{b_1^w, ..., b_n^w, ...b_N^w\}$ for workers and $\{b_m^r\}_{m=1}^M$ from requesters, if worker $w_n$ wins the auction by bidding $b_n^w$ or $b_n^{w\prime}$, then reward $p_n$ to $w_n$ is the same for both.

*Proof.* Lemmas 3 and 4 can be proved by McAfee double auction, which is detailed in Section II-A. □

With the help of the above four Lemmas, we now prove that TODA-SS achieves cost-truthfulness for both requesters and workers.

**(1) Cost-truthfulness for requesters**

We first prove that TODA-SS enforces cost-truthfulness for requesters, that is, no requesters can obtain higher utility through bidding $b_m^{r\prime} \neq v_m^r$. The discussion can be divided into the following three cases.

1) CASE 1: $b_m^{r\prime} = 0$. If requester $r_m$ abandons bidding or his/her bid is 0, he/she cannot be selected as a winner and his/her utility $u_{r,m}^{\text{SS}}$ is 0.
2) CASE 2: $b_m^{r\prime} < v_m^r$. If requester $r_m$ is selected as a winner by the platform, his/her utility $u_{r,m}^{\text{SS}}$ is $v_m^r - q_m$ according to equation (3). We know that $v_m^r$ is a constant as it is the true value of the task submitted by requester $r_m$. Also, $q_m$ is a constant according to Lemmas 1 and 3. Hence, the utility $u_{r,m}^{\text{SS}}$ of $r_m$ is fixed. In addition, if $b_m^{r\prime}$ is too small, requester $r_m$ may not be selected as a winner, and thus his/her utility $u_{r,m}^{\text{SS}}$ is 0. Therefore, the utility $u_{r,m}^{\text{SS}}$ of $r_m$ is not more than that of $b_m^{r\prime} = v_m^r$.
3) CASE 3: $b_m^{r\prime} > v_m^r$. If requester $r_m$ is selected as a winner by the platform, his/her utility $u_{r,m}^{\text{SS}}$ is still $v_m^r - q_m$. More importantly, the situation of $q_m > v_m^r$ (i.e., the utility of $r_m$ is negative) will not happen as our proposed TODA-SS mechanism satisfies individual rationality (the details are shown in Theorem 3). In addition, if requester $r_m$ is not selected as a winner by the platform, his/her utility $u_{r,m}^{\text{SS}}$ is 0. Therefore, the utility $u_{r,m}^{\text{SS}}$ of $r_m$ is not more than that of $b_m^{r\prime} = v_m^r$.

In summary, TODA-SS achieves cost-truthfulness for requesters.

**(2) Cost-truthfulness for workers**

Next, we prove that TODA-SS satisfies cost-truthfulness for workers, that is, no workers can obtain a higher utility through bidding $b_n^{w\prime} \neq c_n^w$. The discussion can be divided into the following three cases.

1) CASE 1: $b_n^{w\prime} = 0$. If worker $w_n$ abandons bidding or his/her bid is very large, he/she cannot be selected as a winner and his/her utility $u_{w,n}^{\text{SS}}$ is 0.
2) CASE 2: $b_n^{w\prime} > c_n^w$. If worker $w_n$ is selected as a winner by the platform, his/her utility $u_{w,n}^{\text{SS}}$ is $p_n - c_n^w$ according to equation (4). We know that $c_n^w$ is a constant as it is the true cost for completing the task submitted by worker $w_n$. Also, $p_n$ is a constant according to Lemmas 2 and 4. Hence, the utility $u_{w,n}^{\text{SS}}$ of $w_n$ is fixed. In addition, if $b_n^{w\prime}$ is too large, worker $w_n$ may not be selected as a winner, and then his/her utility $u_{w,n}^{\text{SS}}$ is 0. Therefore, the utility $u_{w,n}^{\text{SS}}$ of $w_n$ is not more than that of $b_n^{w\prime} = c_n^w$.
3) CASE 3: $b_n^{w\prime} > c_n^w$. If worker $w_n$ is selected as a winner by the platform, his/her utility $u_{w,n}^{\text{SS}}$ is still $p_n - c_n^w$. More importantly, the situation of $p_n < c_n^w$ (i.e., the utility of $w_n$ is negative) will not happen as our proposed TODA-SS mechanism satisfies individual rationality (the details are shown in Theorem 3). In addition, if worker $w_n$ is not selected as a winner by the platform, his/her utility $u_{w,n}^{\text{SS}}$ is 0. Therefore, the utility $u_{w,n}^{\text{SS}}$ of $w_n$ is not more than that of $b_n^{w\prime} = c_n^w$.

In summary, TODA-SS achieves cost-truthfulness for workers.

From the above descriptions, we show that no users can improve their utility by bidding untruthfully, thus our proposed TODA-SS mechanism satisfies cost-truthfulness. □

**Theorem 3.** *TODA-SS satisfies individual rationality.*

*Proof.* To prove the individual rationality of TODA-SS, we need to prove that for each requester $r_m$ ($r_m \in R$), his/her utility $u_{r,m}^{\text{SS}} \geq 0$ and for each worker $w_n$ ($w_n \in W$), his/her utility $u_{w,n}^{\text{SS}} \geq 0$.

**(1) Individual rationality for requesters**

For requester $r_m$, the pricing (i.e., fee) for $r_m$ in each slot is the product of task $\tau_k$'s bidding threshold $\delta_R^k$ and specific demand $\lambda^k$ of task $\tau_k$. If $r_m$ is not selected as a winner by the platform, his/her utility $u^{\text{SS}}_{r,m}$ is 0. If $r_m$ is selected as a winner by the platform and his/her specific demand of task $\tau_k$ is not fully completed, i.e., $\lambda^k_m > \lambda^k_R$, his/her fee $q_m$ is given by

$$q_m = q^k_m = \lambda^k_R * \delta^k_R < \lambda^k_m * \delta^k_R \leq b^r_m \leq v^r_m. \tag{9}$$

If $r_m$ is selected as a winner by the platform and his/her specific demand of task $\tau_k$ is fully completed, his/her fee $q_m$ is given by

$$q_m = q^k_m = \lambda^k_m * \delta^k_R \leq b^r_m \leq v^r_m. \tag{10}$$

Then, we have

$$u^{\text{SS}}_{r,m} = v^r_m - q_m \geq 0. \tag{11}$$

**(2) Individual rationality for workers**

For worker $w_n$, the pricing (i.e., reward) for $w_n$ in each slot is the bidding threshold of task $\tau_k$. If $w_n$ is not selected as a winner by the platform, his/her utility $u^{\text{SS}}_{w,n}$ is 0, otherwise, his/her reward $p_n$ is given by

$$p_n = p^k_n = \delta^k_W \geq b^w_n \geq c^w_n. \tag{12}$$

Then, we have

$$u^{\text{SS}}_{r,m} = p_n - c^w_n \geq 0. \tag{13}$$

In summary, the proposed TODA-SS mechanism satisfies individual rationality. □

**Theorem 4.** *TODA-SS is budget balanced.*

*Proof.* At each slot, TODA-SS collects the fees from requesters as the budget for selecting workers at the next slot. From steps 13-14 in Algorithm 1, it is guaranteed that the reward for each worker winner does not exceed current slot-budget $U^k_t$. Therefore, the total payment to all winners of workers will not exceed the fees collected from requesters. In summary, TODA-SS is budget balanced. □

**Theorem 5.** *TODA-SS satisfies consumer sovereignty.*

*Proof.* In TODA-SS, users are not automatically rejected and are selected as winners as long as their bids meet the bidding thresholds and the slot-budget is not exhausted. Hence, users compete fairly in TODA-SS. In addition, TODA-SS is an online mechanism that allows continuous bidding. For a user who is rejected in a slot, he/she can also participate in the task auction at the next slot. If his/her bid meets the new thresholds at the new slot, the platform will select him as a winner. In summary, TODA-SS satisfies consumer sovereignty. □

### B. TODA-SM and TODA-MM Auction Mechanisms

When a requester recently has multiple tasks with the same (different) type(s) and urgency degree or a worker recently has a single (multiple) task skill(s) and relatively long online time, the user will adopt the OSM (OMM) model and TODA-SS (TODA-OMM) auction mechanism to quickly achieve task allocation and auction bidding.

As discussed in Section III-B, the difference between the OSM model and the OSS model is the number of tasks in each bid, and the difference between the OMM model and the OSM model is the number of submitted bids. Similar to the TODA-SS mechanism, the TODA-SM and TODA-MM mechanisms also consist of three aspects: winner selection, task pricing and threshold updating. Table II shows the most critical indicator of winner selection and threshold updating in three proposed online double auction mechanisms. Each element in Table II is the average bid of each task $\tau_k$ in the corresponding mechanism. Table III shows the most critical indicator of task pricing in three proposed online auction mechanisms.

TABLE II: Critical indicators for winner selection and threshold updating

| TODA-SS | TODA-SM | TODA-MM |
|---|---|---|
| $\frac{b^r_m}{\lambda^k_m}\,(\tau_k \in J_m)$ | $\frac{b^r_m}{\sum_k \lambda^k_m}\,(\tau_k \in J_m)$ | $\frac{b^r_m}{\sum_k \lambda^k_m}\left(\tau_k \in J^{(i)}_m\right)$ |
| $b^k_n\,(\tau_k \in \Gamma_n)$ | $\frac{b^k_n}{\lvert\Gamma_n\rvert}\,(\tau_k \in \Gamma_n)$ | $\frac{b^k_n}{\lvert\Gamma^{(i)}_n\rvert}\left(\tau_k \in \Gamma^{(i)}_n\right)$ |

TABLE III: Critical indicators for task pricing

| TODA-SS | TODA-SM | TODA-MM |
|---|---|---|
| $q_m = q^k_m$ | $q_m = \sum_k q^k_m$ | $q_{m,i} = \sum_k q^k_{m,i}$; $q_m = \sum\limits_m q_{m,i}$ |
| $p_n = p^k_n$ | $p_n = \sum_k p^k_n$ | $p_{n,i} = \sum_k p^k_{n,i}$; $p_n = \sum\limits_n p_{n,i}$ |

In the proposed TODA-SM mechanism, for requester $r_m$, the bid of task $\tau_k\,(\tau_k \in J_m)$ is the average bid $b^r_m/\sum_k \lambda^k_m$, and the pricing of $r_m$ is the total fees of all tasks in the submitted bidding information, i.e., $q_m = \sum_k q^k_m$. For worker $w_n$, the bid of task $\tau_k(\tau_k \in \Gamma_n)$ is the average bid $b^k_n/|\Gamma_n|$, where $|\Gamma_n|$ is the number of tasks in $w_n$'s task set $\Gamma_n$. The pricing of $w_n$ is the total rewards of all tasks in the submitted bidding information, i.e., $p_n = \sum_k p^k_n$.

The TODA-MM mechanism is a generalization of the TODA-SM mechanism. In the proposed TODA-MM mechanism, each user can submit multiple bids to maximize own benefits, and each bid is independent of each other and contains multiple tasks. For a requester/worker, the bid of a task is the average bid of all tasks and the pricing of the requester/worker is the total fees/rewards of all tasks in the submitted bidding information, which are detailed in Tables II and III.

Next, we prove that our proposed TODA-SM and TODA-MM auction mechanisms satisfy the economic properties mentioned in Section II-B.

**Theorem 6.** *TODA-SM and TODA-MM satisfy time-truthfulness.*

*Proof.* The critical difference between TODA-SS and TODA-SM is the number of tasks in the submitted bidding information. The critical difference between TODA-SM and TODA-MM is the number of bids in the submitted bidding information. As with the TODA-SS mechanism, the submitted bidding information in the TODA-SM and the TODA-MM mechanisms also contain arrival and departure time. Therefore, the time-truthfulness of users in the TODA-SM and the

TODA-MM mechanisms can be proved similarly to the time-truthfulness of users in TODA-SS (refer to Theorem 1). □

**Theorem 7.** *TODA-SM and TODA-MM satisfy cost-truthfulness.*

*Proof.* In TODA-SM and TODA-MM, we perform winner selection, task pricing and threshold updating by using M-cAfee double auction, which are the same as TODA-SS. Thus, the cost-truthfulness of users in the TODA-SM and the TODA-MM mechanisms can be proved similarly to the cost-truthfulness of users in TODA-SS (refer to Theorem 2). □

**Theorem 8.** *TODA-SM and TODA-MM satisfy individual rationality.*

*Proof.* To prove the individual rationality of TODA-SM, we need to prove that for each requester $r_m$ ($r_m \in R$), his/her utility $u^{\text{SM}}_{r,m} \geq 0$ and for each worker $w_n$ ($w_n \in W$), his/her utility $u^{\text{SM}}_{w,n} \geq 0$.

**(1) Individual rationality for requesters**

For requester $r_m$, $b^r_m$ is the bid of $r_m$ for tasks in $J_m$ and $v^r_m$ is the true value of these tasks to $r_m$. The relationship between $b^r_m$ and $v^r_m$ is $b^r_m \leq v^r_m$. The fee for $r_m$ in each slot is the pricing of all tasks contained in $J_m$. Specifically, it is the sum of the product of each task's bidding threshold $\delta^k_R$ and task's specific demand $\lambda^k$. If $r_m$ is not selected as a winner by the platform, his/her utility $u^{\text{SM}}_{r,m}$ is 0. If $r_m$ is selected as a winner by the platform and his/her specific demands of all tasks are not fully completed, i.e., $\sum_k \lambda^k_m > \sum_k \lambda^k_R$, his/her fee $q_m$ is given by

$$\begin{aligned} q_m = \sum\nolimits_k q^k_m &= \sum\nolimits_k \left(\lambda^k_R * \delta^k_R\right) \\ &< \sum\nolimits_k \left(\lambda^k_m * \delta^k_R\right) \leq b^r_m \leq v^r_m. \end{aligned} \tag{14}$$

If $r_m$ is selected as a winner by the platform and his/her specific demands of all tasks are fully completed, his/her fee $q_m$ is given by

$$q_m = \sum\nolimits_k q^k_m = \sum\nolimits_k \left(\lambda^k_m * \delta^k_R\right) \leq b^r_m \leq v^r_m. \tag{15}$$

Based on (14) and (15), we have $v^r_m \geq q_m$. Then, the utility of $r_m$ is

$$u^{\text{SM}}_{r,m} = v^r_m - q_m \geq 0. \tag{16}$$

Therefore, $u^{\text{SM}}_{r,m}$ is non-negative, and then TODA-SM satisfies individual rationality for requesters.

**(2) Individual rationality for workers**

For worker $w_n$, $b^w_n$ is the bid of $w_n$ for tasks in $\Gamma_n$ and $c^w_n$ is his/her true cost for completing these tasks. The relationship between $b^w_n$ and $c^w_n$ is $b^w_n \geq c^w_n$. the pricing (i.e., reward) for $w_n$ in each slot is the total bidding thresholds of all submitted tasks. If $w_n$ is not selected as a winner by the platform, his/her utility $u^{\text{SM}}_{w,n}$ is 0, otherwise his/her reward $p_n$ is given by

$$p_n = \sum\nolimits_k p^k_n = \sum\nolimits_k \delta^k_W \geq \sum\nolimits_k b^w_n(t^k_n) = b^w_n \geq c^w_n, \tag{17}$$

Then, we have $p_n \geq c^w_n$. The utility of $w_n$ is

$$u^{\text{SM}}_{w,n} = p_n - c^w_n \geq 0. \tag{18}$$

Therefore, $u^{\text{SM}}_{w,n}$ is non-negative, and then TODA-SM satisfies individual rationality for workers.

In summary, the proposed TODA-SM mechanism satisfies individual rationality.

In TODA-MM, each user can submit multiple bids and each bid is independent of each other. Each bid can be regarded as an independent bid to auction. Therefore, the individual rationality of users in the TODA-MM mechanism can be proved similarly to the individual rationality of users in the TODA-SM mechanism. □

**Theorem 9.** *TODA-SM and TODA-MM are budget balanced.*

*Proof.* Similar to TODA-SS, TODA-SM and TODA-MM charge the fees from requesters in each slot as the budget for selecting workers in the next slot. When selecting workers, they need to judge whether the slot-budget is exhausted, which is the same as TODA-SS. Therefore, TODA-SM and TODA-MM are budget balanced. □

**Theorem 10.** *TODA-SM and TODA-MM satisfy consumer sovereignty.*

*Proof.* Similar to TODA-SS, in TODA-SM and TODA-MM, users are not automatically rejected but are selected as winners as long as their bids meet the bidding thresholds and the slot-budget is not exhausted. Also, TODA-SM and TODA-MM allow continuous bidding. Therefore, our TODA-SM and TODA-MM mechanisms achieve consumer sovereignty. □

## V. Application Example

In this section, we introduce an example to show how our proposed TODAs work in realistic crowdsourcing applications. Specifically, we consider a crowdsourcing application in IoV environments that can serve three types of tasks: sensing tasks, computational tasks, and survey tasks. Users including task requesters and workers can flexibly select auction service models and corresponding auction mechanisms according to the current interested tasks and online duration. To illustrate this, we present the following three-demand example:

- ***Single-task Demand***

We first consider the single-task demand of users. We take the sensing task as an example, because the sensing tasks are common in IoV. For example, some internet map service (e.g., Baidu Map or High Moral Map) needs to obtain real-time road scenes and traffic information. To illustrate the single-task demands of requesters and workers, we introduce the following two users.

1) Requester 1 has a sensing task, specifically an image capture task $\tau_k$ with a specified region size. His/her task demand $\lambda^k_1$ is abstracted as the number of required task results, that is, the specific number of workers who need to complete the task. The demand is generally greater than one due to individual differences. The urgency degree $\mu^k_1$ corresponds to the tolerance delay of completing the image capture task. To quickly achieve task allocation and auction bidding, requester 1 adopts the OSS model to submit his/her task and corresponding bid. In general, requesters who currently have a task, especially an urgent one, prefer the OSS model rather than the OSM and OMM models. This is

because the OSM and OMM models give priority to matching workers with multi-task demands, which will increase the allocation time of a single task and reduce the allocation efficiency;

2) Worker 2 has a short online duration, and he/she only has a single task skill during the recent online period, specifically the image capture skill for task $\tau_k$. To quickly achieve task allocation and auction bidding, worker 2 adopts the OSS model to submit his/her interested task and corresponding bid. In general, workers with only a single task skill, especially those who have a short online duration, prefer the OSS model rather than the OSM and OMM models. The detailed reason is the same as that of requester 1.

Given the above demand background, at each slot $t$, we assume that there are multiple requesters like requester 1, and multiple workers like worker 1. $\delta_R^k, \delta_W^k, \lambda_R^k, \lambda_W^k$ are the thresholds for image capture task $\tau_k$ updated at the end of the previous slot $t-1$ using Algorithm 3 of TODA-SS, and also the prediction of the users' demands to task $\tau_k$ in the current slot $t$. $\delta_R^k$ and $\delta_W^k$ are the bidding thresholds for task requesters and workers to image capture task $\tau_k$, respectively. Also, $\lambda_R^k$ is the task demand threshold of image capture task $\tau_k$, which is the maximum number of required results among all winning requesters in the previous slot $t-1$. $\lambda_W^k$ is the worker demand threshold of image capture task $\tau_k$, which is the number of winning workers required for capturing all image regions of task $\tau_k$ in the previous slot $t-1$. Based on the above thresholds, we select the winners of requesters and workers in this slot $t$ using Algorithm 1 of TODA-SS, and obtain the fees for winning requesters and the rewards for winning workers using Algorithm 2 of TODA-SS.

- ***Single-type Multiple-task Demand***

To illustrate the single-type multiple-task demands of requesters and workers, we introduce the following two users.

1) Requester 3 has multiple sensing tasks, specifically multiple image capture tasks with different locations and area sizes. These tasks are of similar urgency degree. To quickly achieve task allocation and auction bidding, requester 3 adopts the OSM model instead of OSS and OMM models to submit his/her tasks and corresponding bids. This is because the workers using the OSS model generally have a short online duration, and it is difficult for them to complete multiple tasks during this online period. Also, the OMM model gives priority to matching workers with multi-type rather than single-type task skills. Moreover, if these tasks contain one or more tasks with a higher urgency degree, requester 5 can submit more urgent tasks to the OSS model;
2) Worker 4 has a single task skill during the recent online period, and his/her online duration is relatively long. Specifically, he/she has the image capture skill for task $\tau_k$. To earn more rewards, worker 4 adopts the OSM model instead of OSS and OMM models to submit his/her multiple interested tasks with same type and corresponding bids.

The winner selection, task pricing and threshold updating processes are given by our TODA-SM auction mechanism.

- ***Multiple-type Multiple-task Demand***

We then consider the multiple-type multiple-task demand of users. We take the sensing tasks, computational tasks, and survey tasks as the examples. In realistic IoV environment, computational tasks can be the image processing after capturing the environment images, and survey tasks may be the the survey demands initiated by businesses to investigate the driving preference of different IoV users. To illustrate the multiple-type multiple-task demands of requesters and workers, we introduce the following two users.

1) Requester 5 has multiple tasks with multiple types. In addition, these tasks are of similar urgency degree. To clearly describe each task demands and quickly achieve task allocation, requester 5 adopts the OMM model instead of OSS and OSM models to submit his/her tasks and corresponding bids. This is because the workers who select OSS and OSM model generally have a single task skill, and cannot complete multiple-type tasks. Moreover, if these multi-type tasks contain some tasks with a higher urgency degree, requester 5 can submit the more urgent tasks to the OSS or OSM models;
2) Worker 6 has multiple task skills during the recent online period, and his/her online duration is relatively long. To clearly describe the task skills and earn more rewards, worker 6 adopts the OMM model to submit his/her interested tasks and corresponding bids.

The winner selection, task pricing and threshold updating processes are given by our TODA-MM auction mechanism.

## VI. Simulation Results and Performance Evaluation

In this section, we evaluate the advantages of our TODAs compared to the auction mechanism in [18]. We specifically implement these mechanisms and run extensive tests on a Windows PC with Intel Core I5 and 8GB memory. In [18], the authors investigated online two-sided auction among single-type task requesters and workers by adopting McAfee double auction with a fixed auction service model. To verify our advantages of online two-sided auctions in practical crowdsourcing applications, we thus compare our proposed TODAs with the auction mechanism in [18].

### A. Performance Metrics and Simulation Setup

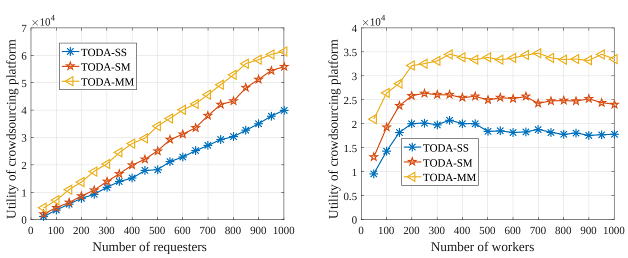
We study the utility of the platform and the average utility of users (i.e., task requesters and workers). Specifically, we evaluate each metric by varying the number of requesters $M$ and the number of workers $N$ from 50 to 1000 with an increment 50, respectively. To evaluate the impact of $M$, we fix $N = 500$. Similarly, to evaluate the impact of $N$, we fix $M = 500$. We set the deadline as 10s and consider 10 slots (i.e., each time slot is 1s). We set the number of types of tasks as 10. The valuation of each task for requesters is uniformly distributed over $(0, 200]$ and the cost of each task for workers is uniformly distributed over $(0, 50]$. In TODA-SS, each user (i.e, requester or worker) can submit one bid containing one task. In TODA-SM, each user can submit one bid containing multiple tasks and the number of tasks does not exceed 5. In TODA-MM, each user can submit up to 5 bids and the number of tasks in each bid does not exceed 5.

### *B. Utility Analysis Based on Simulation Results*

In this subsection, we show the utility performances in our proposed TODA-SS, TODA-SM and TODA-MM auction mechanisms. Moreover, we compare their utility performances with the auction mechanism in [18]. The user utility and platform utility are calculated by equations (3)-(8). Specifically, for each winning requester, the utility is the difference between the true value of the submitted task(s) and the fee charged by the platform. For each winning worker, the utility is the difference between the reward from the platform and the true cost of completing the submitted task(s). For the platform, the utility is the difference between fees from requesters and the rewards to workers. In our TODAs, requesters can select the appropriate auction service model and auction mechanism according to their number of tasks, number of task types, and urgency degree of tasks. Similarly, workers can select the corresponding auction model and mechanism according to their task skills and online time. In the auction mechanism in [18], only requesters requesting simple tasks of a certain type and workers with the corresponding task skill can achieve successful bids during the auction process.

- **Utility of the crowdsourcing platform**

Figs. 6 (a) and (b) show the utility of the crowdsourcing platform with different auction mechanisms versus the number of requesters and the number of workers, respectively.



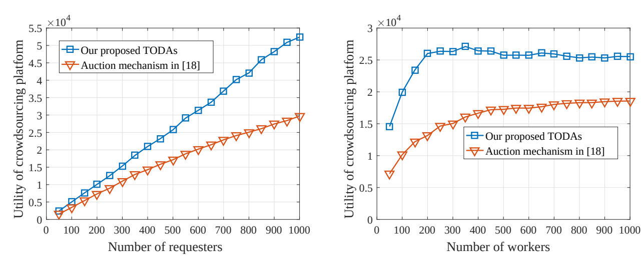

(a) (b)

Fig. 6: The utility of crowdsourcing platform in our proposed TODA-SS, TODA-SM and TODA-MM mechanisms: (a) Platform utility with 500 workers versus the number of requesters, (b) Platform utility with 500 requesters versus the number of workers.

In Fig. 6 (a), we find that the utility of crowdsourcing platform in TODA-SM is higher than that of TODA-SS and less than that of TODA-MM. This is because the users (i.e., requesters and workers) in TODA-SM can submit and complete more tasks than that of TODA-SS, and users in TODA-MM can submit and complete more tasks than that of TODA-SM. More completed tasks lead to higher platform utility. Moreover, in Fig. 6 (a), the platform utility in each auction mechanism increases with the increase of the number of requesters (i.e., $M$), because the workers can complete more tasks when $M$ increases. However, due to the need to allocate multiple tasks with multiple types, TODA-MM has a lower task allocation efficiency and longer task execution time than TODA-SS and TODA-SM. As such, it is not suitable for users with a short online duration. In contrast, for TODA-SS and TODA-SM, although the platform utility of each service process is lower than TODA-MM, they provide higher task allocation efficiency and are more suitable for users with short online durations or a single task type. Moreover, requesters with urgent tasks can quickly allocate tasks and obtain task results by selecting TODA-SS. Therefore, compared with TODA-MM, TODA-SS and TODA-SM can meet users' rapid response requirements.

In Fig. 6 (b), we can find that the platform utility rises and then remains steady (slight decrease) with the increasing number of workers. We next analyze the reasons from two stages. The number of maximum completed tasks is constant due to the fixed number of requesters. Under this setup, in the first stage, more workers lead to more completed tasks, resulting in the increasing platform utility. In the second stage, when the market is saturated with workers, the platform utility almost remains steady. This is because the execution cost of each task is almost constant for workers, and therefore the workers will not continually lower their bids for each task. Moreover, when the x-axis values in Figs. 6 (a) and (b) are less than 500, the platform utility in Fig. 6 (b) is higher than that in Fig. 6 (a) due to more tasks. In contrast, the platform utility in Fig. 6 (b) is lower than that in Fig. 6 (a) due to fewer tasks when the x-axis values are higher than 500.

Figs. 7 (a) and (b) show the utility of the platform in our TODAs and the auction mechanism in [18] changing with the number of requesters and workers, respectively.




(a) (b)

Fig. 7: The utility of crowdsourcing platform in TODAs and the auction mechanism in [18]: (a) Platform utility with 500 workers versus the number of requesters, (b) Platform utility with 500 requesters versus the number of workers.

We observe from Fig. 7 that TODAs always have higher platform utility than the auction mechanism in [18] regardless of the number of task requesters and workers. The reason for this improvement comes from two aspects. First, our TODAs provide an on-demand service strategy, while the auction mechanism in [18] only provides a fixed auction model for servicing a single type of task. Second, TODAs can effectively select workers with the appropriate skills for different tasks, while the auction mechanism in [18] makes an unreasonable assumption that each task can be completed by a single worker. Clearly, the auction mechanism in [18] is impractical since the auction requirements and interested tasks in realistic crowdsourcing applications are extremely diverse, and some complex or urgent tasks may require multiple workers. In contrast, our TODAs can provide specific auction service models for different users with various task requirements and effectively match the demands and supplies of crowdsourcing markets. Through the above analysis, we conclude that the task execution efficiency and quantity of our proposed TODAs are higher than [18], which contribute to the higher platform utility in Fig. 7.

- **Average utility of requesters**

The size relationships of the requester utilities in TODA-SS, TODA-SM and TODA-MM are the same as the platform utilities in Fig. 6. Figs. 8 (a) and (b) show the average utility of requesters in TODAs and the mechanism in [18] changing with the number of requesters and workers, respectively.

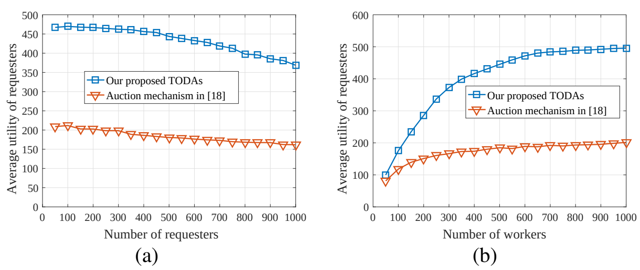



Fig. 8: The average utility of requesters in TODAs and the auction mechanism in [18]: (a) Requester average utility with 500 workers versus the number of requesters, (b) Requester average utility with 500 requesters versus the number of workers.

In Fig. 8 (a), with the increase of the number of requesters (i.e., $M$), the average utilities of requesters in TODAs and the auction mechanism in [18] descend slowly. This is because with more requesters, the competition among requesters becomes more fierce due to the constant number of workers. Then, the platform generally selects requesters with higher bids as the auction winners. This leads to an average increase in the fees charged to requesters. In Fig. 8 (b), with the increase of the number of workers (i.e., $N$), the average utilities of requesters in our proposed TODAs mechanisms and the auction mechanism in [18] increase and then gradually tend to be stable. The reason is that the number of requesters is constant and more workers lead to more completed tasks. This then creases the average utility of requesters to increase. When the market stabilizes, the average utility of requesters will remain steady. Another observation from Fig. 8 is that TODAs perform better than the auction mechanism in [18] regardless of the number of task requesters and workers. This is because the auction mechanism in [18] can only serve a single type of simple task, and thus requesters with other types of tasks or complex tasks cannot successfully bid. Here, complex tasks refer to tasks that require multiple workers to complete together. In contrast, our proposed TODAs provide three requirement-based online auction service models and corresponding auction mechanisms, which can accommodate diversified tasks and bidding demands for different requesters.

- **Average utility of workers**

The size relationships of the worker utilities in TODA-SS, TODA-SM and TODA-MM are the same as the platform utilities in Fig. 6. Figs. 9 (a) and (b) show the average utility of workers in TODAs and the mechanism in [18] changing with the number of requesters and workers, respectively.

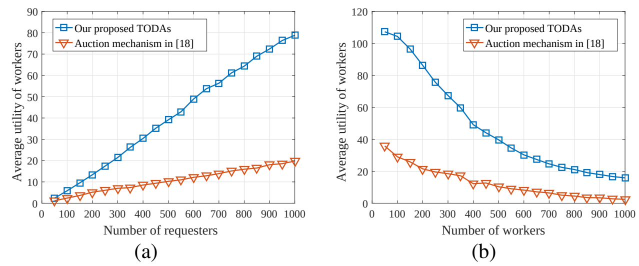



Fig. 9: The average utility of workers in TODAs and the auction mechanism in [18]: (a) Worker average utility with 500 workers versus the number of requesters, (b) Worker average utility with 500 requesters versus the number of workers.

In Fig. 9 (a), with the increase of $M$, the average utilities of workers in TODAs and the auction mechanism in [18] increase. This is because with more requesters, the workers can complete more tasks and earn more rewards. In Fig. 9 (b), with the increase of $N$, the average utilities of workers drop dramatically in both TODAs and the auction mechanism in [18]. The reason is that the number of tasks is constant, and with more workers, the competition among workers becomes more fierce and it leads to a decrement in payments for workers. Moreover, we find from Fig. 9 that TODAs perform better than the auction mechanism in [18] regardless of the number of task requesters and workers. The detailed reason is the same as that of the requester average utility in Fig. 8.

### *C. Truthfulness Analysis Based on Simulation Results*

Next, we evaluate the truthfulness of our proposed TODA-SS, TODA-SM and TODA-MM auction mechanisms in terms of time-truthfulness and cost-truthfulness. Since the proposed TODAs are similar in terms of winner selecting, task pricing and threshold updating, we only prove the truthfulness of TODA-SS to save space (once TODA-SS is proven, TODA-SM and TODA-MM are also proven).

- **Time-truthfulness**

For each user, the submitted arrival and departure time in the submitted bidding information are $at$ and $dt$, respectively. Besides, $\overline{at}$ and $\overline{dt}$ are respectively his/her true arrival and departure time. We first verify the time-truthfulness of TODA-SS by randomly picking a requester and a worker and allowing them to submit their arrival/departure time that are different from their true arrival/departure time. We illustrate the results in Fig. 10.

In Figs. 10 (a) and (b), we can see that the requester and the worker achieve their optimal utility if they submit the true arrival time (i.e., $\overline{at} = at = 30$). We next analyze the simulation results from two stages. In the first stage, the arrival time submitted by the requester or the worker is lower than his/her actual arrival time, which is obviously untruthful

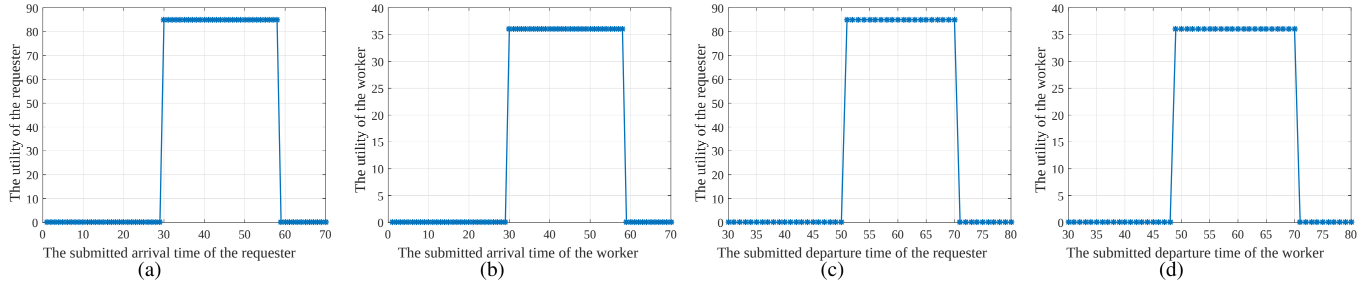



Fig. 10: The true arrival and departure time of the user are $\overline{at} = 30$ and $\overline{bt} = 70$, respectively: (a) The utility of the selected requester with $bt = 70$ versus his/her submitted arrival time $at$, (b) The utility of selected worker with $bt = 70$ versus his/her submitted arrival time $at$, (c) The utility of selected requester with $at = 30$ versus his/her submitted departure time $dt$, (d) The utility of selected worker with $at = 30$ versus his/her submitted departure time $dt$.

behavior. Their utilities in this stage are zero because they are offline and cannot send/receive tasks to/from the platform. When $at = \overline{at} = 30$, they submit the true arrival time in the biding information and obtain the optimal utility due to being chosen to be the winners. After that, in the second stage, their submitted arrival time is higher than the actual arrival time. In fact, a typical user will not choose this behavior because it is not conducive to improving his/her utility. Specifically, submitting a delayed arrival time may cause the platform to determine that the user cannot complete the task or wait for the task to complete because his/her online time is short. As such, he/she will not be selected as a winner. This is why the user utility in Figs. 10 (a) and (b) drops to 0 when the submitted arrival time is about 60 instead of 70.

In Figs. 10 (c) and (d), we see that the requester and the worker achieve their optimal utility if they submit the true departure time (i.e., $\overline{dt} = dt = 70$). The explanation for this is similar to Figs. 10 (a) and (b). Through the above analysis, we conclude that submitting any untruthful arrival and departure time does not improve the utility of the requester and worker. Therefore, our proposed TODA-SS mechanism achieves time-truthfulness.

- **Cost-truthfulness**

Next, we verify the cost-truthfulness of TODA-SS by randomly picking a requester and a worker and allowing them to submit bids that are different from their true value and cost for the submitted tasks. We illustrate the results in Fig. 11.

In Fig. 11 (a), we can see that the requester achieves his/her optimal utility if he/she bids truthfully (i.e., $b_m = v_m = 150$). We next analyze the simulation results. Due to the individual rationality for task requesters, the submitted bid for the task is not higher than the true value of the task. When the bid is lower than the bid threshold (current market value) for requesters, this requester will not be selected as the winner, and then his/her utility is zero. Otherwise, he/she will become a winner, and the utility is constant as the difference between the true value and the bidding threshold for requesters.

In Fig. 11 (b), it can be found that the worker achieves his/her optimal utility if he/she bids truthfully (i.e., $b_n = c_n = 50$). The explanation for this is similar to Fig. 11 (a). This shows that submitting any untruthful bids does not improve the utility of the requester and worker, and thus our proposed TODA-SS achieves cost-truthfulness.

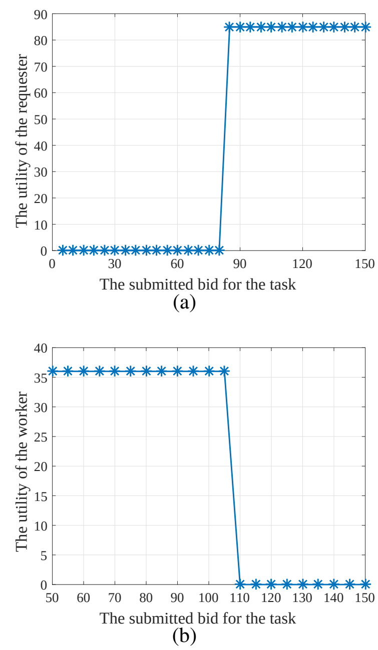



Fig. 11: (a) The utility of selected requester versus his/her submitted bid for the task (the true value of the task is 150), (b) The utility of selected worker versus his/her submitted bid of the task (the true cost of the task is 50).

## VII. CONCLUSION

An effective crowdsourcing application requires the participation of a large number of users comprising both task requesters and workers. Meanwhile, it must provide various task requirements for different users. Online double auctions are proven paradigms to stimulate and serve users in mobile crowdsourcing. Unfortunately, the related works mainly concentrate on designing two-sided auction mechanisms for single-type tasks using fixed auction service models, which are impractical because the demands and supplies in realistic crowdsourcing markets are extremely diverse. To this end, we focus on an on-demand service strategy, and then design three online service models called OSS, OSM and OMM models. Furthermore, by adopting McAfee double auction, we propose three truthful online double auction mechanisms for three service models, namely TODA-SS, TODA-SM and TODA-MM mechanisms. Based on these, users can select the appropriate auction service model and corresponding auction mechanism according to diversified tasks and bidding demands. Finally, we conduct extensive theoretical proofs and simulation experiments, and it is verified that our proposed three TODAs satisfy time-truthfulness, cost-truthfulness, individual rationality, budget balance and consumer sovereignty. Moreover, we show that our TODAs mechanisms can ensure the various demands of different users and improve platform utility and average user utility.

## References


[1] B. Cao, S. Xia, J. Han and Y. Li, "A Distributed Game Methodology for Crowdsensing in Uncertain Wireless Scenario," *IEEE Trans. Mobile Comput.*, vol. 19, no. 1, pp. 15-28, Jan. 2020.

[2] X. Cao, P. Yang, F. Lyu, J. Han, Y. Li, D. Guo and X. Shen, "Trajectory Penetration Characterization for Efficient Vehicle Selection in HD Map Crowdsourcing," *IEEE Internet Things J.*, vol.8, no. 6, pp. 4526-4539, Mar. 2021.

[3] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun and A. Manzoor and C. S. Hong, "A Crowdsourcing Framework for On-Device Federated Learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3241-3256, Feb. 2020.

[4] Y. Yu, S. Liu, L. Guo, P. L. Yeoh, B. Vucetic and Y. Li, "CrowdR-FBC: A Distributed Fog-Blockchains for Mobile Crowdsourcing Reputation Management," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8722-8735, May. 2020.

[5] L. Zhang, B. Cao, Y. Li, M. Peng and G, Fang, "A Multi-Stage Stochastic Programming based Offloading Policy for Fog Enabled IoT-eHealth" *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 411-425, Apr. 2020.

[6] A. Hamrouni, H. Ghazzai, M. Frikha and Y. Massoud, "A Spatial Mobile Crowdsourcing Framework for Event Reporting," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 2, pp. 477-491, Apr. 2020.

[7] Y. Wang, W. Dai, Q. Jin and J. Ma. "BciNet: A Biased Contest-Based Crowdsourcing Incentive Mechanism Through Exploiting Social Networks," *IEEE Trans. Syst., Man, Cybern.*, vol. 6, no. 2, pp. 1-12, Jun. 2018.

[8] Y. Liu, H, Wang, M. Peng, J. Guan and Y. Wang, "An Incentive Mechanism for Privacy-Preserving Crowdsensing via Deep Reinforcement Learning", *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8616-8631, 2021.

[9] Y. Liu, X. Xu, J. Pan and J. Zhang, "A Truthful Auction Mechanism for Mobile Crowd Sensing With Budget Constraint," *IEEE Access*, vol. 7, no. 1, pp. 43933-43947, Mar. 2019.

[10] R. Zhou, Z. Li and C. Wu, "A Truthful Online Mechanism for Location-Aware Tasks in Mobile Crowd Sensing," *IEEE Trans. Mobile Comput.*, vol. 17, no. 8, pp. 1737-1749, Nov. 2018.

[11] D. Zhao, X. Li and H. Ma, "How to Crowdsource Tasks Truthfully without Sacrificing Utility: Online Incentive Mechanisms with Budget Constraint," in *Proc. INFOCOM*, Toronto, ON, Canada, pp. 1213-1221, May. 2014.

[12] D. Zhao, H. Ma and X. Ji, "Generalized Lottery Trees: Budget-Balanced Incentive Tree Mechanisms for Crowdsourcing," *IEEE Trans. Mobile Comput.*, doi: 10.1109/TMC.2020.2979459, Mar. 2020.

[13] C. Tang, X. Li, M. Cao, Z. Zhang and X. Yu, "Incentive Mechanism for Macrotasking Crowdsourcing: A Zero-Determinant Strategy Approach," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8589-8601, Oct. 2019.

[14] X. Wang, W. Tushar, C. Yuen and X. Zhang, "Promoting Users' Participation in Mobile Crowdsourcing: A Distributed Truthful Incentive Mechanism (DTIM) Approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5570-5582, May. 2020.

[15] C. Liu, S. Wang, L. Ma, X. Cheng, R. Bie and J. Yu, "Mechanism Design Games for Thwarting Malicious Behavior in Crowdsourcing Applications," in *Proc. IEEE INFOCOM*, Atlanta, GA, USA, pp. 1-9, Oct. 2017.

[16] X. Zhang, G. Xue, R. Yu, D. Yang and J. Tang, "Countermeasures Against False-Name Attacks on Truthful Incentive Mechanisms for Crowdsourcing," *IEEE J. Sel. Areas in Commun.*, vol. 35, no. 2, pp. 478-485, Jan. 2017.

[17] Y. Liu, T. Feng, M. Peng, J. Guan and Y. Wang, "DREAM: Online Control Mechanisms for Data Aggregation Error Minimization in Privacy-Preserving Crowdsensing," *IEEE Trans. Depend. Secure Comput.*. DOI: 10.1109/TDSC.2020.3011679, Jul. 2020.

[18] Y. Wei, Y. Zhu, H. Zhu, Q. Zhang and G. Xue, "Truthful Online Double Auctions for Dynamic Mobile Crowdsourcing," in *Proc. IEEE INFOCOM*, Hong Kong, China, pp. 2074-2082, Apr. 2015.

[19] H. Huang, Y. Xin, Y. E. Sun and W. Yang, "A truthful Double Auction Mechanism for Crowdsensing Systems with Max-min Fairness," in *Proc. IEEE WCNC*, San Francisco, CA, USA, pp. 1-6, Mar. 2017.

[20] X. Zhang, L. Gao, B. Cao, Z. Li and M. Wang, "A Double Auction Mechanism for Mobile Crowd Sensing with Data Reuse," in *Proc. IEEE GLOBECOM*, Singapore, pp. 1-6, Dec.2017.

[21] W. Jin, M. Li, L. Guoy and L. Yang, "DPDA: A Differentially Private Double Auction Scheme for Mobile Crowd Sensing," in *Proc. CNS*, Beijing, China, pp. 1-9, May. 2018.

[22] J. Shu, K. Yang, X. Jia, X. Liu, C. Wang and R. Deng, "Proxy-Free Privacy-Preserving Task Matching with Efficient Revocation in Crowdsourcing," *IEEE Trans. Depend. Secure Comput.*, vol. 18, no. 1, pp. 117-130, Jan. 2021.

[23] S. Chen, M. Liu and X. Chen, "A Truthful Double Auction for Two-sided Heterogeneous Mobile Crowdsensing Markets," *Computer Communications*, vol. 81, pp. 31-42, 2016.

[24] Y. Tong, Y. Zeng, B. Ding, L. Wang and L. Chen, "Two-Sided Online Micro-task Assignment in Spatial Crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 5, pp. 2295-2309, May. 2021.

[25] Y. Lin, Z. Cai, X. Wang and F. Hao, "Incentive Mechanisms for Crowdblocking Rumors in Mobile Social Networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 9220-9232, Jul. 2019.

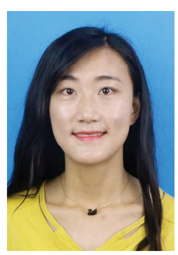
[26] J. Lin, M. Li, D. Yang and G. Xue, "Sybil-Proof Online Incentive Mechanisms for Crowdsensing," in *Proc. IEEE INFOCOM*, Honolulu, HI, USA, pp. 2088-2096, Apr. 2018.

[27] G. Li and J. Cai, "An Online Incentive Mechanism for Crowdsensing with Random Task Arrivals," *IEEE Internet Things J.*, vol 7, no. 4, pp. 2982-2995, Jan. 2020.

[28] X. Zhou and H. Zheng, "TRUST: A General Framework for Truthful Double Spectrum Auctions," in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, pp. 999-1007, Apr. 2009.

[29] R. P. McAfee, "A Dominant Strategy Double Auction," *J. Econ. Theory*, vol. 56, no. 2, pp. 434-450, Apr. 1992.

[30] D. Zhao, X. Y. Li and H. Ma, "Budget-Feasible Online Incentive Mechanisms for Crowdsourcing Tasks Truthfully," *IEEE/ACM Trans. Networking*, vol. 24, no. 2, pp. 647-661, Apr. 2016.

**Shumei Liu** received the B.S. degree in electronic and information engineering from Shanxi University, Taiyuan, China, in 2016 and the M.S. degree in electronics and communication engineering from Northeastern University, Shenyang, China, in 2018. She is currently pursuing the Ph.D. degree in communication and information system at Northeastern University. Her research interests include the Internet-of-Things (IoT), mobile edge computing, and radio resource management for enhancing the physical layer security. She has received the Best Paper Award in National Postdoctoral Academic Forum in China (2018).

**Yao Yu** (Member, IEEE) received B.S. degree in communication engineering from the Northeastern University, Shenyang, China in 2005, and the Ph.D. degree in communication and information system from the Northeastern University, Shenyang, China in 2010. From 2010 to 2011, she was a Postdoctoral Fellow with Department of Computing at Hong Kong Polytechnic University, Hong Kong, China. Also she was a visiting scholar in the University of Sydney from 2019 to 2020. She is currently a Professor at the School of Computer Science and Engineering, Northeastern University, Shenyang, China. Her current research interests include network security and big data. She is a member of the IEEE.

**Lei Guo** (Senior Member, IEEE) received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2006. He is currently a Full Professor with Chongqing University of Posts and Telecommunications, Chongqing, China. He has authored or coauthored more than 200 technical papers in international journals and conferences. He is an Editor for several international journals. His current research interests include communication networks, optical communications, and wireless communications.

**Phee Lep Yeoh** (Member, IEEE) received the B.E. degree with University Medal and the Ph.D. degree from the University of Sydney (USYD), Australia, in 2004 and 2012, respectively. From 2005 to 2008, he was a Wireless Technology Specialist at Telstra, Australia. From 2012 to 2016, he was a Lecturer and Research Fellow in Wireless Communications at the University of Melbourne, Australia. Since 2016, he has been a Senior Lecturer with the School of Electrical and Information Engineering at USYD. His research interests include secure communications for the Internet-of-Things (IoT), ultra-reliable and low-latency communications (URLLC), and multi-scale molecular communications.

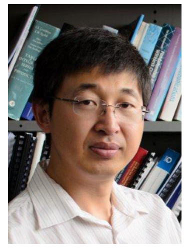
Dr. Yeoh is a recipient of the 2020 USYD Robinson Fellowship, the 2018 Alexander von Humboldt Research Fellowship for Experienced Researchers, and the 2014 Australian Research Council (ARC) Discovery Early Career Researcher Award (DECRA). He has received best paper awards at IEEE ICC 2014 and IEEE VTC-Spring 2013, and best student paper awards with his supervised students at the 2013 and 2019 Australian Communications Theory Workshop (AusCTW).

**Qiang Ni** (Senior Member, IEEE) is a Professor at the School of Computing and Communications, Lancaster University, Lancaster, U.K. His research interests include the area of future generation communications and networking, including green communications and networking, millimeter-wave wireless communications, cognitive radio network systems, non-orthogonal multiple access (NOMA), heterogeneous networks, 5G and 6G, SDN, cloud networks, edge computing, dispersed computing, energy harvesting, wireless information and power transfer, IoTs, cyber physical systems, AI and machine learning, big data analytics, and vehicular networks. He has authored or co-authored 300+ papers in these areas. He was an IEEE 802.11 Wireless Standard Working Group Voting Member and a contributor to various IEEE wireless standards.

**Branka Vucetic** (Life Fellow, IEEE) received the Ph.D. degree from the University of Belgrade, Belgrade, Serbia, in 1982.

She is an ARC Laureate Fellow and Director of the Centre of Excellence for IoT and Telecommunications at the University of Sydney. Her current research work is in wireless networks and the Internet of Things. In the area of wireless networks, she works on communication system design for millimetre wave frequency bands. In the area of the Internet of Things, Vucetic works on providing wireless connectivity for mission critical applications.

Prof. Vucetic is a life Fellow of IEEE, the Australian Academy of Technological Sciences and Engineering and the Australian Academy of Science.



**Yonghui Li** (Fellow, IEEE) received his Ph.D. degree from Beijing University of Aeronautics and Astronautics, Beijing, China, in November 2002. From 1999-2003, he was affiliated with Linkair Communication Inc, where he held a position of project manager with responsibility for the design of physical layer solutions for the LAS-CDMA system. Since 2003, he has been with the Centre of Excellence in Telecommunications, the University of Sydney, Australia. He is now a Professor in School of Electrical and Information Engineering, University of Sydney. He is the recipient of the Australian Queen Elizabeth II Fellowship in 2008 and the Australian Future Fellowship in 2012. His current research interests are in the area of wireless communications, with a particular focus on MIMO, millimeter wave communications, machine to machine communications, coding techniques and cooperative communications. He holds a number of patents granted and pending in these fields.

Prof. Li is a Fellow of IEEE. He is now an editor for IEEE Transactions on Communications and IEEE Transactions on Vehicular Technology. He also served as a guest editor for several special issues of IEEE journals, such as IEEE JSAC special issue on Millimeter Wave Communications. He received the best paper awards from IEEE International Conference on Communications (ICC) 2014, IEEE PIMRC 2017 and IEEE Wireless Days Conferences (WD) 2014.