

University of Mississippi

eGrove

Faculty and Student Publications

Engineering, School of

1-1-2020

TIMCC: On Data Freshness in Privacy-Preserving Incentive Mechanism Design for Continuous Crowdsensing Using Reverse Auction

Xiaoqiang Ma

Huazhong University of Science and Technology

Weiwei Deng

Huazhong University of Science and Technology

Feng Wang

University of Mississippi

Menglan Hu

Huazhong University of Science and Technology

Fei Chen

Qingdao University

Follow this and additional works at: https://egrove.olemiss.edu/engineering_facpubs



next page for additional authors
part of the [Biomedical Engineering and Bioengineering Commons](#), [Chemical Engineering Commons](#), [Civil and Environmental Engineering Commons](#), [Computer Engineering Commons](#), [Electrical and Computer Engineering Commons](#), [Geological Engineering Commons](#), and the [Mechanical Engineering Commons](#)

Recommended Citation

Ma, X., Deng, W., Wang, F., Hu, M., Chen, F., & Hassan, M. M. (2020). TIMCC: On Data Freshness in Privacy-Preserving Incentive Mechanism Design for Continuous Crowdsensing Using Reverse Auction. *IEEE Access*, 8, 1777–1789. <https://doi.org/10.1109/ACCESS.2019.2962212>

This Article is brought to you for free and open access by the Engineering, School of at eGrove. It has been accepted for inclusion in Faculty and Student Publications by an authorized administrator of eGrove. For more information, please contact egrove@olemiss.edu.

Authors

Xiaoqiang Ma, Weiwei Deng, Feng Wang, Menglan Hu, Fei Chen, and Mohammad Mehedi Hassan

Received November 22, 2019, accepted December 18, 2019, date of publication December 25, 2019, date of current version January 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962212

TIMCC: On Data Freshness in Privacy-Preserving Incentive Mechanism Design for Continuous Crowdsensing Using Reverse Auction

XIAOQIANG MA¹, (Member, IEEE), WEIWEI DENG¹,
FENG WANG², (Senior Member, IEEE), MENGLAN HU¹, FEI CHEN³,
AND MOHAMMAD MEHEDI HASSAN⁴, (Senior Member, IEEE)

¹School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China

²Department of Computer and Information Science, The University of Mississippi, Oxford, MS 38677, USA

³College of Computer Science and Technology, Qingdao University, Qingdao 266071, China

⁴College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

Corresponding author: Mohammad Mehedi Hassan (mmhassan@ksu.edu.sa)

This work was supported by King Saud University, Riyadh, Saudi Arabia, through Researchers Supporting Project number RSP-2019/18.

ABSTRACT As an emerging paradigm that leverages the wisdom and efforts of the crowd, mobile crowdsensing has shown its great potential to collect distributed data. The crowd may incur such costs and risks as energy consumption, memory consumption, and privacy leakage when performing various tasks, so they may not be willing to participate in crowdsensing tasks unless they are well-paid. Hence, a proper privacy-preserving incentive mechanism is of great significance to motivate users to join, which has attracted a lot of research efforts. Most of the existing works regard tasks as one-shot tasks, which may not work very well for the type of tasks that requires continuous monitoring, e.g., WIFI signal sensing, where the WIFI signal may vary over time, and users are required to contribute continuous efforts. The incentive mechanism for continuous crowdsensing has yet to be investigated, where the corresponding tasks need continuous efforts of users, and the freshness of the sensed data is very important. In this paper, we design *TIMCC*, a privacy-preserving incentive mechanism for continuous crowdsensing. In contrast to most existing studies that treat tasks as one-shot tasks, we consider the tasks that require users to contribute continuous efforts, where the freshness of data is a key factor impacting the value of data, which further determines the rewards. We introduce a metric named *age of data* that is defined as the amount of time elapsed since the generation of the data to capture the freshness of data. We adopt the reverse auction framework to model the connection between the platform and the users. We prove that the proposed mechanism satisfies individual rationality, computational efficiency, and truthfulness. Simulation results further validate our theoretical analysis and the effectiveness of the proposed mechanism.

INDEX TERMS Crowdsensing, incentive mechanism, privacy-preserving, age of data.

I. INTRODUCTION

Recent years have witnessed the proliferation of smartphones and wearable devices [1]. Thanks to the various functionalities of embedded sensors, these devices not only have plenty of computation and communication capabilities, but also become an important interface between users and the environment [2]. With potential users round-the-clock, large area coverage, and low maintenance cost,

crowdsensing — *crowdsourcing with mobile devices*, a paradigm that makes use of the wisdom and efforts of the crowd, has shown its great impact in many areas, e.g., indoor localization [3], [4], environmental monitoring [5], [6], traffic prediction [7], and healthcare [8].

It has been reported that crowdsensing applications have attracted millions of contributors [9]. Performing crowdsensing tasks, however, will inevitably incur some costs to users such as energy consumption and possible privacy leakage. Therefore, the users may not be willing to join the paradigm unless they are well-paid. Without enough users

The associate editor coordinating the review of this manuscript and approving it for publication was Shui Yu.

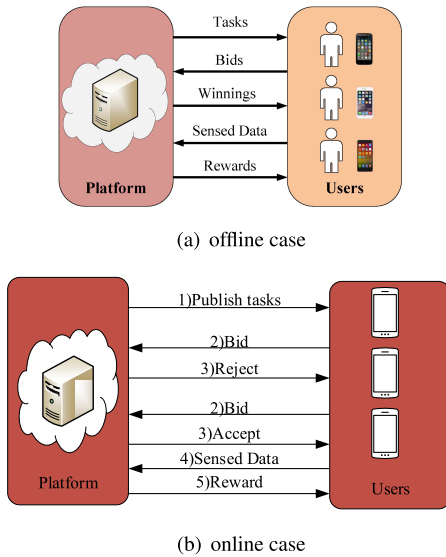


FIGURE 1. An illustrative example of a crowdsensing auction model.

to participate, the corresponding tasks may not be completed successfully. Therefore, a proper incentive mechanism that provides appropriate payment to users is of great significance to the success of crowdsensing [10], [11].

Many research efforts have been poured into the design of incentive mechanisms for crowdsensing [11], among which reverse auction is one of the most commonly adopted models [12], [13]. The basic idea of reverse auction is presented in Fig. 1. The users act as the seller, while the platform acts as the buyer. The platform (a.k.a. auctioneer) first allocates tasks. The users then submit to the platform a bid set containing one or more tasks that they are interested in, along with the bid price for processing the tasks. Based on the users' bids, the platform determines the winners as well as the payments according to their bids. Finally, the winners execute the tasks and receive the rewards from the platform. The situation can be divided into offline case and online case. For the offline case, the platform (a.k.a. auctioneer) selects the winners when it collects all the bids, while for the online case, the bids come in the form of flow, and the platform must make a decision (accept or reject) when a bid comes.

Although most of the existing solutions regard tasks as *one-shot* tasks (like “go and fetch the ball”), most crowdsensing tasks such as monitoring the traffic condition of a town need the continuous contribution of users, because the traffic condition varies over the time, making most of the previous incentive mechanisms can not work very well in such situations. Another shortcoming is that most of the existing solutions neglect the time-sensitivity, i.e., they do not care about when the users contribute to the system. But in continuous crowdsensing, the freshness of data is very important, especially in the presence of a monitoring task, which urges the incentive mechanism to take the freshness of data into consideration.

Inspired by [14]–[17], we introduce a metric *age of data* which is defined as the amount of time elapsed since the

generation of the data. Naturally, it captures the freshness of data. We adopt the reverse auction framework to simulate the connection between the platform and users. We take the metric age of data into consideration, and make it an influencing factor of the value of data, which further influences the rewards. Furthermore, we design an incentive mechanism **TIMCC** (Truthful Incentive Mechanism for Continuous Crowdsensing) for both offline and online cases. TIMCC runs round by round and assumes the age of data is negatively correlated with the value of data, and then the value of data influences the payment for winners. In each round, TIMCC selects the winning bids first, and then determines the rewards for the winners.

Our main contributions are summarized as following:

- To the best of our knowledge, our work is the first to design truthful incentive mechanism for continuous crowdsensing, taking into consideration the relationship between the value of data and its age.
- We combine the reverse auction framework with the age of data and propose an incentive mechanism TIMCC where a bid's value is related to its age.
- We prove that TIMCC satisfies individual rationality, computational efficiency, and truthfulness, which is also validated by extensive simulations.

The rest of this paper is organized as follow. In Section II, we formulate the system model along with the definition of the age of data. In Section III, we introduce the strategic behaviors, several properties, and the detail of TIMCC. In Section IV, we provide the theoretical analysis of TIMCC. Simulation results are presented in Section V, and some related works are briefly reviewed in Section VI. We discuss some future directions in Section VII, and conclude this paper in Section VIII.

II. SYSTEM MODEL

A. CROWDSENSING AND AUCTION MODEL

Table 1 summarizes the notations used in this paper. We consider a discrete-time system with time slots $t = 1, 2, \dots, T$. There is a platform resided in the cloud and a job Γ to be completed. Participators connect with the cloud through some well-designed networks [18], [19]. The job can be divided into many indivisible small tasks $\Gamma = \{\tau_1, \tau_2, \dots, \tau_m\}$. Small task τ_i needs q_i (may be different between small tasks) users to contribute to it at each slot. Then for all small tasks, we have a vector $\mathcal{Q} = (q_1, q_2, \dots, q_m)$. To a certain extent, q_i reflects the importance degree of small task τ_i . For example, if $q_1 = 3$ and $q_2 = 4$, then at each slot, small task τ_1 needs three users and τ_2 needs four users. τ_2 is attached to more significance than τ_1 . In addition, we consider the situation where all small tasks need the continuous contribution of users. For example, a job to draw a map of the traffic condition of a town. The whole area contains many points of interests. Each point is associated with a small task. The point with higher vehicle flow may have a higher q . Furthermore, as the traffic condition varies over time, continuous update is needed while

TABLE 1. Summary of notations.

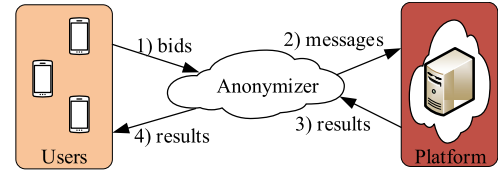
Symbol	Description
$\mathcal{P} = \{p_1, p_2, \dots, p_n\}$	set of users
$\Gamma = \{\tau_1, \tau_2, \dots, \tau_m\}$	set of small tasks
$B_j^k = (\sigma_j^k, t_j^k, a_j^k)$	user p_j 's k -th bid
$R_j^k(t)$	reward for bid B_j^k at time slot t
$R(t)$	reward vector at time slot t
$A_j^k(t)$	age of bid B_j^k at time slot t
U_j^k	utility of user p_j 's k -th bid
T	maximum number of time slots
K	age threshold
q_i	number of users that task τ_i needs
x_j^k	times that bid B_j^k wins
σ_j^k	small task of bid B_j^k
t_j^k	bid time of bid B_j^k
a_j^k	bid price of bid B_j^k
$v_j^k(t)$	value of bid B_j^k at time slot t
$\mathcal{V}_j^k(A)$	value of bid B_j^k when its age is A

the map is drawn. Users are thus required to make continuous efforts to draw the map and update it. In this case, the map needs to be updated once a slot with the sensed data. Such a job Γ is completed if and only if, at all time slots, all small tasks in Γ are completed.

There is a set of users $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$ who can provide services to collect data to contribute to the job. Consider a user cannot appear at two different places at the same time slot, we make an assumption that each user $p_j \in \mathcal{P}$ can only contribute to one small task at a slot, but can contribute to different small tasks at different slots. Contributing to the corresponding task will incur a cost to the user such as resource consuming and privacy leakage. The cost is private and only known to the user itself.

Considering that many crowdsensing applications and websites have attracted millions of clicks, competition exists among users. We use the *reverse auction* framework to model the competition and the interaction between the platform and users. The reason why we call it reverse auction is that the role of seller and buyer are reversed in the auction. Users are sellers and sell their services, the platform serves as the buyer and buys service of users. The platform selects some users as winners according to some mechanisms, and only winners have the opportunity to provide services and receive the reward.

Due to the distributed nature of crowdsensing, users may suffer the privacy leakage when submit their bids. There are many researches that focus on the privacy protection. The authors in [20] use the k -anonymity to protect user's privacy, and they employ a trusted third-party (anonymizer). Users first send their message to the anonymizer which then aggregates the message of k users within a cloaked region and sends the cloaked message to the platform. We adopt this manner, and assume that there is a trusted third-party (anonymizer). Users do not connect with the platform directly, instead they submit their bids to the anonymizer first, and then the

**FIGURE 2.** An illustrative of anonymizer.

anonymizer aggregates the message and sends to the platform. Fig. 2 gives an illustration of how the anonymizer works.

Then the crowdsensing works as follow:

1) *Tasks Publication*: The platform publishes the job, along with the corresponding set of small tasks, the vector $\mathcal{Q} = (q_1, q_2, \dots, q_m)$.

2) *Bids Submission*: Each user p_j submits its m_j bids to the anonymizer: $(\sigma_j^1, t_j^1, a_j^1), (\sigma_j^2, t_j^2, a_j^2), \dots, (\sigma_j^{m_j}, t_j^{m_j}, a_j^{m_j})$, where σ_j^k is a small task that user p_j is willing to contribute to; t_j^k denotes a time slot when user p_j can contribute its efforts, i.e., when it can collect the sensed data; a_j^k is the bid price, which means that user p_j will refuse to accept the corresponding task if it fails to receive a reward no less than a_j^k . We use c_j^k to denote the true cost of user p_j 's contributing to the task σ_j^k . Note that a_j^k is not always equals to c_j^k . Consider that a single user can not appear at two different locations at the same time, we set a constraint that each user can submit no more than one bid in a time slot. Therefore, m_j has an upper bound T (which is the maximum number of time slots), i.e., $m_j \leq T$ for all j . We summarize the m_j bids in a vector $B_j = (B_j^1, B_j^2, \dots, B_j^{m_j})$, where B_j^k is the k -th bid of user p_j , i.e., $B_j^k = (\sigma_j^k, t_j^k, a_j^k)$. We further summarize the bids of all users in a vector $B = (B_1, B_2, \dots, B_n)$. The anonymizer aggregates all the bids and send the aggregated queries to the platform.

3) *Winning Bids Selection*: After receiving all the message, the platform selects the winning bids and allocates the tasks according to some specific mechanisms.

4) *Tasks Execution*: The winners contribute to the tasks and send the sensed data to the platform.

5) *Receiving Payment*: Platform sends the payment to the anonymizer, and then the anonymizer sends the rewards to the corresponding users. Assume the reward bid B_j^k obtained at slot t is $R_j^k(t)$ ($R_j^k(t) = 0$ if bid B_j^k loses at slot t), and then the total reward bid B_j^k obtained is the sum of $R_j^k(t)$, i.e., $R_j^k = \sum_{t=1}^T R_j^k(t)$. Accordingly, the reward user p_j obtained is the total reward of its all m_j bids: $R_j = \sum_{k=1}^{m_j} R_j^k = \sum_{k=1}^{m_j} \sum_{t=1}^T R_j^k(t)$. We summarize two reward vectors: $R = (R_1, R_2, \dots, R_n)$ and $R(t) = (R_1(t), R_2(t), \dots, R_n(t))$, where R denotes the total rewards of all users, and $R(t)$ denotes the rewards at time slot t of all users.

B. AGE OF DATA

Now we introduce the definition of age of data. Inspired by [14], [16], [21], we define the age of data as the amount

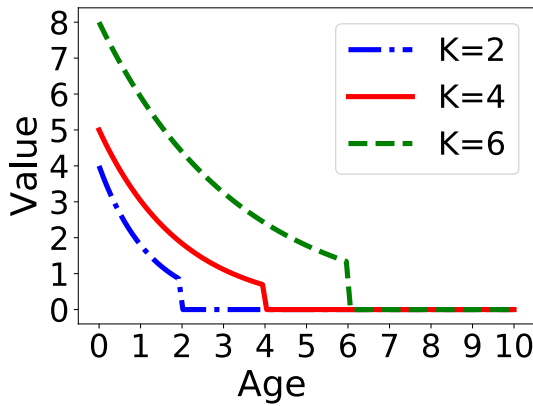


FIGURE 3. Relationship between age and value.

of time elapsed since the generation of the data. Naturally, it captures the freshness of data. It is equal to zero at the slot when the data is sensed. When we refer to the word “data”, it means the data collected by a specific bid. In this sense, it is equivalent between the age of data and the age of bid. We thus use these two terms interchangeably for convenience. The age of user p_j 's k -th bid B_j^k at slot t is defined as $A_j^k(t) = t - t_j^k$, where $t \geq t_j^k$, and t_j^k is the time slot when the data is collected.

We assume that the age of data has an impact on the value of data. How to measure the value of data is not a simple question [22]. In this paper, we do not investigate this question further and just assume that the value of data can be determined by its age directly. We use $\mathcal{V}_j^k(A)$ to denote the value of data collected by user p_j 's k -th bid B_j^k when its age is A . $\mathcal{V}_j^k(0)$ denotes the initial value. The initial value can be different between users, which is reasonable because different users may be equipped with different devices, or have a different reputation among the platform. If a user holds a more advanced device or has a higher reputation, then it has a higher initial value. We assume that the initial value is known to the platform.

In this paper, we mainly discuss the situation where the value of data is negatively correlated with its age, i.e., $\mathcal{V}_j^k(A_1) < \mathcal{V}_j^k(A_2)$ when $A_1 > A_2$ for every j and k . Furthermore, we set a threshold age K such that $\mathcal{V}_j^k(A) = 0$, when $A \geq K$; $\mathcal{V}_j^k(A) > 0$, when $A < K$. Fig. 3 provides an illustration. The threshold K indicates that when a bid's age is equal to or greater than K , the bid will lose its value and has no chance to win. For the same reason, we also ignore the difference between the value of data and the value of bids for convenience.

III. TIMCC DESIGN

In this section, we first discuss strategic behavior of users in our TIMCC design in subsection III-A and some desired properties in subsection III-B. In subsection III-C, we formulate the details about the design of TIMCC.

A. STRATEGIC BEHAVIOR

Due to the distributed nature of crowdsensing, the real cost c_j^k of user p_j 's performing the small task σ_j^k is private and unknown to the platform. Each user is regarded as a strategic but selfish individual who always wants to maximize its utility, which is the sum of utilities of all its bids. The utility of a bid is the difference between the payment of the bid and the total real cost of performing the corresponding task. We allow a bid to win more than once, and we have an indicator vector $x = (x_1, x_2, \dots, x_n)$ where $x_j = (x_j^1, x_j^2, \dots, x_j^{m_j})$, and x_j^k denotes the times that bid B_j^k wins, it is zero if bid B_j^k loses all the time slots. The task in bid B_j^k is accomplished by user p_j for x_j^k times, for each accomplishment, we assume it occurs a cost c_j^k to the corresponding user. So the utility of user p_j can be computed as

$$U_j = \sum_{k=1}^{m_j} U_j^k \quad (1)$$

$$U_j^k = R_j^k - x_j^k c_j^k = \sum_{t=1}^T R_j^k(t) - x_j^k c_j^k \quad (2)$$

where U_j^k is the utility of bid B_j^k , $R_j^k(t)$ is the reward for bid B_j^k at slot t , $x_j^k c_j^k$ is the total cost of contributing to the corresponding task.

Because each user is regarded as a strategic but selfish individual who always wants to maximize its utility. It is possible that some selfish individuals misreport its costs to maximize its utility, i.e., for some j or k , a_j^k is not equal to c_j^k . Such kind of behavior is called strategic behavior. The existence of strategic behaviors makes it difficult for the platform to select those users with lower real cost.

B. DESIRED PROPERTIES

A well-designed incentive mechanism plays an important role in the success of crowdsensing, where the following three properties are desired.

- **Individual Rationality:** An incentive mechanism is individually rational if a user can always acquire a non-negative utility when reporting its real cost. Individual rationality ensures each user's cost is covered, as the platform fails to attract enough users if its incentive mechanism does not satisfy individual rationality.
- **Computational Efficiency:** An incentive mechanism is computationally efficient if it can be executed within a polynomial time. Computational efficiency is needed to guarantee that the incentive mechanism is practically executable.
- **Truthfulness:** An incentive mechanism is truthful if for each user, it cannot obtain a higher utility by misreporting its cost. Truthfulness is designed to make the incentive mechanism invulnerable to malicious price manipulation. If truthfulness is not satisfied, dishonest users submit untruthful bids and receive a higher reward than they deserve, which damage the interest of honest users.

Among the three properties, truthfulness is the most challenging when designing an incentive mechanism for crowdsensing. Although there are many auctions satisfying truthfulness [12], [23], their truthfulness may not be easily held if simply combined with the concept of the age of data and extended to a multi-round auction. For example, assume π is a truthful auction model, based on which a user p_j submits its truthful bid and gains a reward R_j . Now if it knows the distribution of other bids, it submits a bid with a higher bid price than its truthful bid. It misses the current time slot, but wins at the next slot and then receives a reward R_j^* . Because R_j and R_j^* are calculated based on different bid sets (at different time slots), R_j^* may be larger than R_j . Then user p_j receives a higher reward by submitting untruthful bids, and thus truthfulness is not guaranteed.

Note that we have a threshold age K . When a bid's age is greater than K , its value decreases to zero, and then the bid will have no chance to win. If we set $K = 1$, then all old bids will not be selected. This situation falls to the tradition cases where the age of data is not taken into consideration, and the platform only selects the fresh data [11], [13].

C. DETAILS OF TIMCC

We assume there exists a trust third-party (anonymizer) between users and the platform. The anonymizer collects the bids from users and then aggregates the message of users within a cloaked region to send to the platform. The platform adopts a similar idea with the $k - th$ lowest price auction, in which bidders who submit the $k - 1$ lowest prices are selected as winners, and they are rewarded with the payment of the $k - th$ lowest price. Naturally, the reward is no less than their bid price, and all winners are rewarded the same amount of payment. It has been proven that the auction is truthful [24].

TIMCC runs round by round and allows a bid to win for more than once. For each accomplishment, we assume it occurs a cost to the corresponding user. For the offline case, each round consists of three components. The first component adopts a greedy policy, with the aim of maximizing the ratio of value to cost to determine the winning bids, we call it *Winning Bids Selection*, which is described in Algorithms 1. The second component determines an appropriate payment for each winning bid, and we call it *Pricing* as described in Algorithms 2. The third component updates the bid set and makes preparation for the next round. Consider the third component is very simply, we do not describe it in a separate algorithm and write it into the final algorithm directly.

For each small task $\tau_i \in \Gamma$, $q_i > 0$ winners are required. Winning bids are selected not merely based on the bid price, but also the ratio of value to bid price to achieve the highest gain per unit cost. In this case, bids with the q_i highest ratio of value to bid price are selected as winning bids. The corresponding payment is determined by the *critical ratio*, which is defined as the $q_i + 1 - th$ highest ratio of value to bid price. For ease of presentation, we define $v_j^k(t) = \mathcal{V}_j^k(A_j^k(t))$, the notation $v_j^k(t)$ denotes the value of bid B_j^k at time slot t .

Algorithm 1 WinBidSelection

Input: $B, \Gamma, \mathcal{Q}, t$
Output: B_w, \bar{R}^t

```

1 Initialization:  $B_w \leftarrow \emptyset, \bar{R}^t \leftarrow \vec{0}$ 
2 for each task  $\tau_i \in \Gamma$  do
3    $l \leftarrow 0$ 
4   while  $l < q_i$  do
5      $B_{j^*}^k \leftarrow \arg \max_{B_j^k} \{ \frac{v_j^k(t)}{a_j^k} \mid \sigma_j^k = \tau_i, B_j^k \notin B_w \}$ ;
6      $B_w \leftarrow B_w \cup \{B_{j^*}^k\}$ ;
7      $l \leftarrow l + 1$ ;
8   end
9    $\bar{R}_i^t \leftarrow \max_{B_j^k} \{ \frac{v_j^k(t)}{a_j^k} \mid \sigma_j^k = \tau_i, B_j^k \notin B_w \}$ ;
10 end
11 return  $\bar{R}^t, B_w$ 

```

Algorithm 2 Pricing

Input: B_w, \bar{R}^t, t
Output: $R(t)$

```

1 Initialization:  $R(t) \leftarrow \vec{0}$ 
2 for each winning bid  $B_j^k \in B_w$  do
3    $R_j(t) \leftarrow R_j(t) + \frac{v_j^k(t)}{\bar{R}_i^t}$ 
4 end
5 return  $R(t)$ 

```

Algorithm 1 determines the winning bids, which takes the set of bids B , \mathcal{Q} , the set of small tasks Γ , and the time slot t as input, and outputs the winning bids, as well as the critical ratio vector $\bar{R}^t = (\bar{R}_1^t, \bar{R}_2^t, \dots, \bar{R}_m^t)$, where \bar{R}_j^t is the *critical ratio* for small task τ_j at slot t . Given the time slot, a small task is one-to-one corresponded with its critical ratio. In line 1, we initialize B_w as an empty set and \bar{R}^t as a zero vector. Then the for loop determines the winning bids for each small task. In the for loop, l depicts the number of winners that have been selected. From line 4 to line 8, we select q_i winning bids with the q_i highest ratio of value to bid price for each small task in Γ . In line 9, we compute the $q_i + 1 - th$ highest ratio of value to bid price (critical ratio) \bar{R}_i^t , which will be used to compute the final payment.

Algorithm 2 calculates the payment for each user, which takes the critical ratio vector \bar{R}^t , the set of winning bids B_w , and the time slot t as input, and outputs payment vector $R(t) = (R_1(t), R_2(t), \dots, R_n(t))$, where $R_j(t)$ denotes the payment for user p_j at slot t . If user p_j loses, then its reward $R_j(t) = 0$; otherwise, $R_j(t)$ is set to the corresponding value of bid divided by the critical ratio. We will prove in next section that the payment is no less than the corresponding real cost if user p_j submits its real cost.

Fig. 4 provides an illustration of how TIMCC runs offline, which takes the set of bids B , \mathcal{Q} , the set of small tasks Γ , and

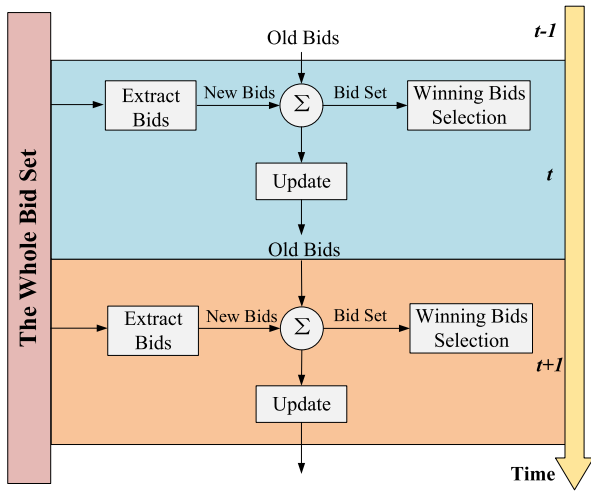


FIGURE 4. An illustration of how TIMCC works.

Algorithm 3 TIMCC-offline

Input: B, Q, Γ, T
Output: R

```

1  $R \leftarrow \bar{0}, B_{old} \leftarrow \emptyset$ 
2 for time slot  $t = 1$  to  $T$  do
3    $B_{new} = \{B_j^k | B_j^k \in B, t_j^k = t\}$ 
4    $B_s = B_{old} \cup B_{new}$ 
5    $(\bar{R}^t, B_w) \leftarrow \text{WinBidSelection}(B_s, \Gamma, Q, t)$ 
6    $R(t) \leftarrow \text{Pricing}(B_w, \bar{R}^t, t)$ 
7    $R \leftarrow R + R(t)$ 
8   for each bid  $B_{j^*}^{k^*}$  in  $B_s$  do
9     update its value to  $v_{j^*}^{k^*}(t+1)$ 
10    if  $v_{j^*}^{k^*}(t+1) \leq 0$  then
11      delete  $B_{j^*}^{k^*}$  from  $B_s$ 
12    end
13  end
14   $B_{old} \leftarrow B_s$ 
15 end
16 return  $R$ 

```

the maximum number of time slots T as input. For each time slot, we extract new bids from the whole bid set, and merge it with the old bid set to be the input of *WinBidSelection* that determines the winning bids and the critical ratios. Accordingly, *Pricing* is run to determine the payment for users. Those processes are detailed in lines 3-7 in Algorithm 3. Finally, the bid set is updated, the value of bids are updated, and bids whose value have decreased to zero are excluded. The updated bid set plays the role of the old bid set that will be used at the next slot.

For the online case, we assume that the bids come in the form of flow in each slot, and the platform must make a decision when it receives a bid. Contrary to the offline case, the platform cannot determine the *critical ratio* based on all the bids, instead it maintains the critical ratio dynamically and

Algorithm 4 TIMCC-online

Input: B, Q, Γ, T
Output: R

```

1  $R \leftarrow \bar{0}, B_{old} \leftarrow \emptyset$ 
2 for time slot  $t = 1$  to  $T$  do
3    $B_{win} \leftarrow \emptyset$ 
4   for each task in  $\tau_i \in \Gamma$  do
5     initialize critical ratio  $r_i$  empirically
6      $v \leftarrow 0, c \leftarrow 0, l \leftarrow 0$ 
7     for each bid  $B_j^k$  aim for task  $\tau_i$  do
8       if  $\frac{v_j^k(t)}{d_j^k} \geq r_i$  then
9          $B_{win} = B_{win} \cup \{B_j^k\}$ 
10         $R_j^k(t) = \frac{v_j^k(t)}{r_i}$ 
11         $v \leftarrow v + v_j^k(t)$ 
12         $c \leftarrow c + d_j^k$ 
13         $r_i \leftarrow \frac{v}{c}$ 
14         $l \leftarrow l + 1$ 
15        if  $l = q_i$  then
16          break
17        end
18      end
19    end
20    if  $l < q_i$  then
21      select winners from  $B_{old}$ 
22    end
23     $B_{old} = B_{old} \cup B_{win}$ 
24    for each bid  $B_{j^*}^{k^*}$  in  $B_{old}$  do
25      update its value to  $v_{j^*}^{k^*}(t+1)$ 
26      if  $v_{j^*}^{k^*}(t+1) \leq 0$  then
27        delete  $B_{j^*}^{k^*}$  from  $B_{old}$ 
28      end
29    end
30  end
31 end
32 return  $R$ 

```

selects bids whose ratio is no less than the critical ratio. The critical ratio is dynamically updated based on the winning bids. Algorithm 4 depicts how TIMCC runs online. Line 5 initializes the critical ratio for the task. When a bid comes, if its ratio is no less than the critical ratio, it is selected as a winning bid immediately. Line 10 determines the reward for the winning bid, line 11-13 update the critical ratios dynamically. We will prove that the critical ratio is updated with no decrease in the next part of this paper, which motivates users to submit their bids earlier. If enough winning bids are selected, the corresponding task will reject all bids that come afterward as line 15-17. If not enough users are selected, the platform turns to the old bids in line 20-22. Finally the old bids is updated in line 23-29, which makes a preparation for the next time slot.

IV. PROPERTIES OF TIMCC

In this section, we analyze TIMCC theoretically to prove that TIMCC satisfies individual rationality, computational efficiency, and truthfulness. We analyze the online TIMCC and prove that the critical ratio is updated with no decrease.

Lemma 1: TIMCC is individually rational.

Proof: Let p_j be a user who bids truthfully, i.e., $a_j^k = c_j^k$. We first prove that a user's reward is no less than her/his cost.

If p_j is a loser for all time slot, then naturally its reward and cost are zero in both offline and online cases.

If p_j is a winner, let its bid B_j^k wins at time slot t . We shall prove that $R_j^k(t) \geq c_j^k$.

For the offline case, from Algorithm 2, we know $R_j^k(t) = \frac{v_j^k(t)}{\bar{R}_j^t}$. Remark that the critical ratio \bar{R}_j^t is no more than the ratio

(value to bid price) of any winning bid, i.e., $\bar{R}_j^t \leq \frac{v_j^k(t)}{a_j^k}$. Hence we have $R_j^k(t) \geq a_j^k = c_j^k$.

For the online case, from line 7 in Algorithm 4, we have $\frac{v_j^k(t)}{a_j^k} \geq r_i$, $R_j^k(t) = \frac{v_j^k(t)}{r_i} \geq a_j^k = c_j^k$.

Then we have TIMCC is individually rational in both offline and online cases. ■

Lemma 2: TIMCC is computationally efficient.

Proof: For the offline case, TIMCC consists of three parts, WinBidSelection, Pricing, and Bids update. The time complexity of WinBidSelection is $O(|\Gamma||B|^2)$, and once WinBidSelection is finished, critical ratios are determined, Pricing and Bids update have lower time complexity (no more than $O(|B|)$). Therefore, WinBidSelection plays a dominant role in TIMCC. TIMCC runs WinBidSelection for T times, $|B| = O(n)$, $|\Gamma| = m$, so the time complexity of offline case is $O(Tmn^2)$.

For the online case, TIMCC selects winning bids dynamically. In each round, TIMCC just traverses the bid set once to select winners and maintain the critical ratio, its time complexity is $O(T|B|) = O(Tn)$. ■

Lemma 3: TIMCC is truthful.

Proof: Assume a user p_j , if it submits its truthful bid $A_j = (\tau_j^k, a_j^k, t_j^k)$, where $a_j^k = c_j^k$, the corresponding utility is denoted as U_j^k . Now if the user submits an untruthful bid $\hat{A}_j = (\tau_j^k, \hat{a}_j^k, t_j^k)$ instead, where $\hat{a}_j^k \neq c_j^k$, and the corresponding utility is \hat{U}_j^k . To prove truthfulness, we shall prove $U_j^k \geq \hat{U}_j^k$.

For the offline case, we decompose U_j^k and \hat{U}_j^k into $U_j^k = \sum_{t=1}^T U_j^k(t)$ and $\hat{U}_j^k = \sum_{t=1}^T \hat{U}_j^k(t)$, where $U_j^k(t)$ and $\hat{U}_j^k(t)$ denote the corresponding utility at the slot t . If we can prove $U_j^k(t) \geq \hat{U}_j^k(t)$ for all t , then truthfulness holds naturally. We have the following case analysis.

Case 1: Both A_j and \hat{A}_j lose at the slot t .

In this case, $U_j^k(t) = 0$, $\hat{U}_j^k(t) = 0$, naturally $U_j^k(t) \geq \hat{U}_j^k(t)$ holds.

Case 2: A_j wins, but \hat{A}_j loses at the slot t .

In this case, $\hat{U}_j^k(t) = 0$. Because A_j is a truthful bid, and it wins, from Lemma 1, we have $U_j^k(t) \geq 0$. So $U_j^k(t) \geq \hat{U}_j^k(t)$ holds.

Case 3: A_j wins, and \hat{A}_j wins too at the slot t .

Remark that in TIMCC, winning bids are selected according to their ratio of value to bid price, assume d winning bids are selected, the reward is determined by the critical ratio which is the $(d + 1) - th$ highest ratio. Because both A_j and \hat{A}_j win, by submitting untruthful bid \hat{A}_j , the critical ratio is not changed, and the reward is not changed further, i.e. $U_j^k(t) = \hat{U}_j^k(t)$. Naturally, $U_j^k(t) \geq \hat{U}_j^k(t)$ holds.

Case 4: A_j loses, but \hat{A}_j wins at the slot t .

For simplicity, we use r_j to denote the critical ratio when truthful bid A_j is submitted, and \hat{r}_j to denote the critical ratio when untruthful bid \hat{A}_j is submitted. Because A_j loses and \hat{A}_j wins, we have $U_j^k = 0$ and $\hat{r}_j \geq r_j$.

Because A_j loses, its ratio is no more than the critical ratio r_j , i.e. $0 < \frac{v}{a_j^k} \leq r_j$, where v denotes the value. Then $c_j^k = a_j^k \geq \frac{v}{r_j}$.

\hat{A}_j wins, its reward at slot t is $R_j^k(t) = \frac{v}{\hat{r}_j} \leq \frac{v}{r_j} \leq c_j^k$. So $\hat{U}_j^k = R_j^k(t) - c_j^k \leq 0$. Then $U_j^k(t) \geq \hat{U}_j^k(t)$ holds.

From the above analysis, we can conclude that offline TIMCC is truthful. Similarly, we can prove that the online TIMCC is truthful. ■

Lemma 4: For the online case, the critical ratio in TIMCC is updated with no decrease.

Proof: Suppose that a bid $B_{j^*}^{k^*}$ wins, whose ratio is $\frac{v_{j^*}^{k^*}}{b_{j^*}^{k^*}}$, and then the critical ratio is updated from $r(t) = \frac{v}{b}$ to $r(t + 1) = \frac{v + v_{j^*}^{k^*}}{b + b_{j^*}^{k^*}}$. We will prove $r(t + 1) \geq r(t)$. Because the bid $B_{j^*}^{k^*}$ wins, we have $\frac{v_{j^*}^{k^*}}{b_{j^*}^{k^*}} \geq r(t)$, then

$$r(t + 1) \geq \frac{b * r(t) + b_{j^*}^{k^*} * r(t)}{b + b_{j^*}^{k^*}} = r(t)$$

V. PERFORMANCE EVALUATION

We conduct extensive simulations to evaluate the performance of TIMCC. We run our tests on a Windows 10 x64 PC with an AMD Ryzen 5 3600 6-Core 3.60 GHz processor and 32 GB memory. Each bid is randomly generated, where the real cost and initial value of each bid follow uniform distribution over $[3, 20]$ and $[1, 20]$, respectively. For the online case, we initialize the critical ratio of all tasks as the ratios of the mean initial value and the mean cost. There are ten time slots $T = 10$, and each bid time is randomly chosen from $\{1, 2, \dots, 10\}$. As for the decrease of value, we mainly focus on the linear case, and the decrease rate is set to 2.5, i.e., when a bid's age increases by 1, its value decreases by 2.5. If there is no special description, m (which is $|\Gamma|$) is fixed to 10, and the age threshold K is fixed to 3. Furthermore,

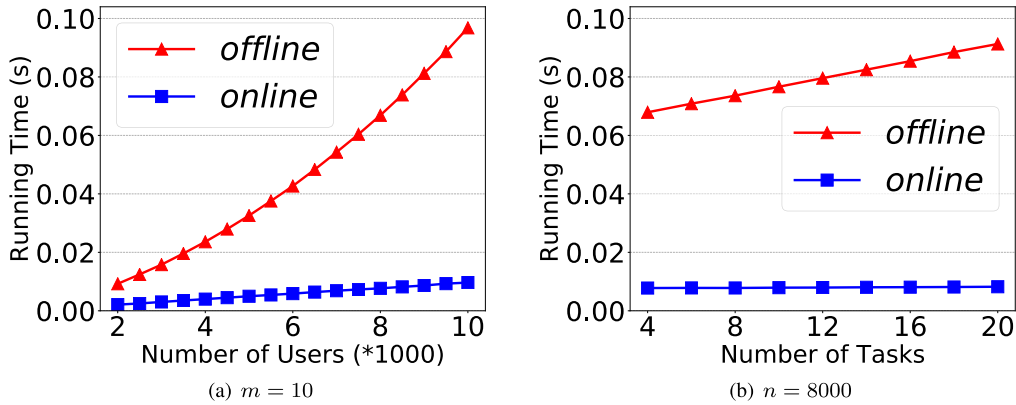


FIGURE 5. Running time.

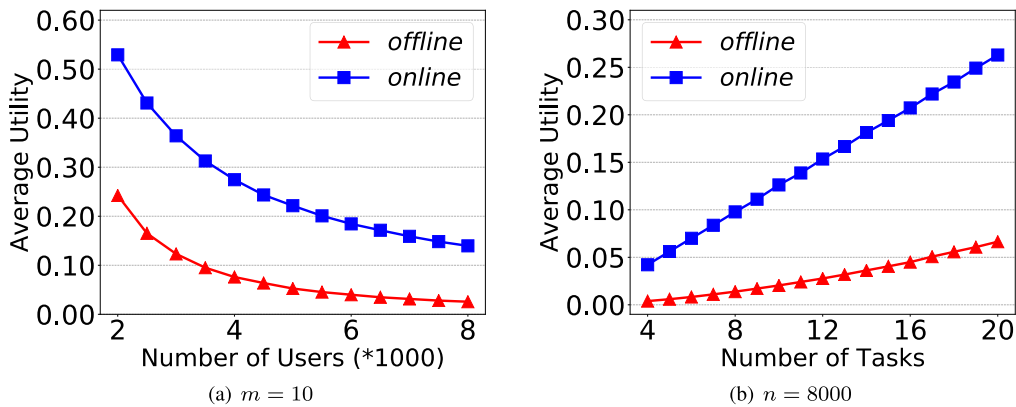


FIGURE 6. Average utility.

we let q_j be a constant, i.e., $q_i = q_j$ for any i and j , which means that different small tasks require the same number of winners. Because for each user p_j , the rewards of the m_j bids are computed independently, we let a user submit only one bid for the clearness of description and comparison. All results are averaged over 100 times for each test.

In the remainder of this section, we first examine the performance of TIMCC. We then further compare TIMCC with IMS-SS [23].

A. PERFORMANCE OF TIMCC

1) RUNNING TIME

Fig. 5 depicts the impact of the number of users n and the number of tasks m on running time. In Fig. 5(a), we set the number of tasks $m = 10$, let n vary from 2000 to 10000 with an increment of 500. For each small task, three winners are required. In Fig. 5(b), we set the number of users n to 8000, let m vary from 4 to 20 with a step of 2. Other parameters are the same as Fig. 5(a).

From Fig. 5(a), we observe that with the increment of n , the running time increases in both the offline and online cases. The offline case increases with a speed of $O(n^2)$, while the online case increases with a linear speed

approximately. From Fig. 5(b), we observe that with the increment of m , the running time of the offline case increases with a linear speed, while the running time of the online case does not increase with m . Those results match our analysis in Lemma 2.

2) AVERAGE UTILITY

Fig. 6 reveals the average user utility, which is defined as the total utility of all users over the number of users. Fig. 6(a) shows the relationship between average utility and the number of users n . The number of users n varies from 2000 to 8000 with a step of 500. We observe that with the increase of n , the average utility drops dramatically in both offline and online cases. This is because if there are more users, the competition among users will become fiercer, thus the platform can select users with lower cost, which will lead to a decrement in payment. Furthermore, although the average utility decrease with n , it never drops to a negative one, which partly matches our analysis in Lemma 1. The reason why we say it partly is that the average utility is non-negative does not mean the utilities of all users are non-negative.

Fig. 6(b) depicts the relationship between the average utility of users and the number of small tasks m . We fix the

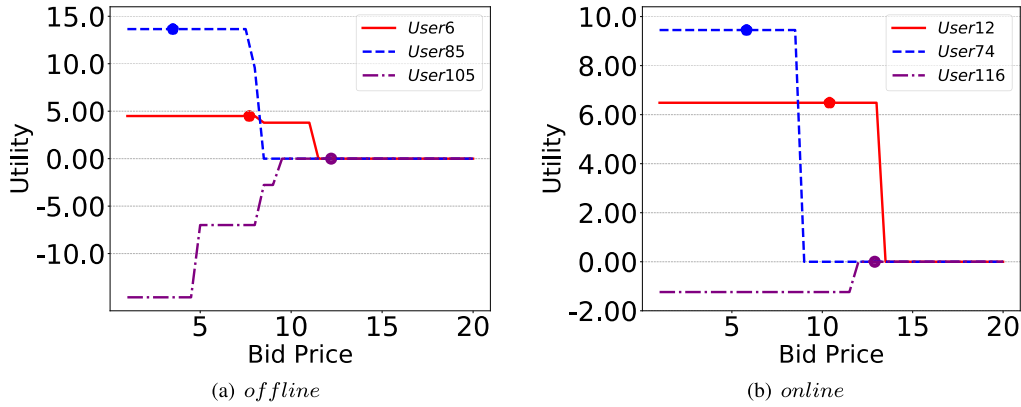


FIGURE 7. Impact of strategic behaviours. (The three cycle dots depict the corresponding truthful bids.)

total number of users n to 8000 and let the number of small tasks vary from 4 to 20 with a step of 2. It is observed that when the number of small tasks increases, the average utility grows in both offline and online cases. On one hand, with more small tasks, more users will be selected as winners and receive the payment. On the other hand, with more small tasks, the competition among users become less fierce, which will lead to a higher reward for winners.

In both Fig. 6(a) and Fig. 6(b), the average utility of offline case is higher than that of online case. It is natural that the platform make a decision on all bids in offline case, so the platform can buy cheaper bids which lead to a lower payment for users.

3) IMPACT OF STRATEGIC BEHAVIORS

The impact of strategic behaviors (untruthful bids) on user utilities is depicted in Fig. 7, where the three circular points denote their truthful bid prices. In this simulation, we set n to 1000. The threshold age is set to 2. In Fig. 7(a), we pick three users: user 6, user 85 and user 105. User 6 and user 85 are winners, while user 105 is a loser when revealing their true costs 7.702, 3.488 and 12.240, respectively. In Fig. 7(b), we pick user 12, user 74 and user 116. User 12 and user 74 are winners, while user 116 is a loser when revealing their true costs 10.402, 5.812 and 12.901, respectively. We let them change their bid prices in $[1.0, 20.0]$ with a step of 0.5. It is observed that no user can receive a higher utility by submitting an untruthful bid in both offline and online cases. Those results match our analysis in Lemma 3.

B. COMPARISON WITH IMS-SS

In this part, we compare the performance of TIMCC (offline) with IMC-SS [23] in terms of *Average utility*, *Gain per payment* and *Probability of failure*. In [23], the authors proposed three different models: single-requester single-bid model (SS-model), single-requester multiple-bid model (SM-model), and multiple-requester multiple-bid model (MM-model). They also designed an incentive mechanism for each of these models.

The reason why we compare TIMCC with IMC-SS (Incentive Mechanism for SS-model) rather than IMC-SM (Incentive Mechanism for SM-model) and IMC-MM (Incentive Mechanism for MM-model) is that in TIMCC, although a user may submit multiple bids, no more than one bid is allowed at a single slot, which is very similar to IMC-SS, where a single user can submit only one bid. IMC-SS first runs a procedure *Job Selection* to select jobs that can be finished with no monopoly user. Then IMC-SS determines winning users according to the minimum ratio of bid price to the number of tasks. Finally, it sets the reward to each winner the critical value. As there exist enough users in our simulation, it is guaranteed that all tasks can be finished without monopoly user. So it is safe to leave out Job Selection in IMC-SS in our scenario.

1) AVERAGE UTILITY & GAIN PER PAYMENT

Note that IMS-SS can be view as a single-round auction, so we fix T in TIMCC to 1. Furthermore, we limit users in IMC-SS to submit bids that contain only one single small task and fix q_i to 1 for all i , i.e., all small tasks in TIMCC only require one winner, then the two models become very similar. In these settings, both TIMCC and IMC-SS require a few winners, so we let the number of users n vary in a relatively small range, and let m vary in a relatively large range.

Fig. 8 depicts how the average utility of users changes when n and m increase in TIMCC and IMS-SS. Fig. 9 depicts the relationship between the gain per payment and n and m in TIMCC and IMS-SS. In Fig. 8(a) and Fig. 9(a), we fix the number of small tasks m to 10, let the number of users n vary from 100 to 400 with a step of 20. In Fig. 8(b) and Fig. 9(b), we fix the number of users n to 2000, and let the number of small tasks m vary from 5 to 100 with a step of 5.

It is observed from Fig. 8 that the average utility in both TIMCC and IMS-SS increases with m , but decreases with n , which matches our previous analysis and the result in [23]. We also find that the average utility of TIMCC is always higher than that of IMC-SS. This is because IMC-SS determines the payment by their bid prices only,

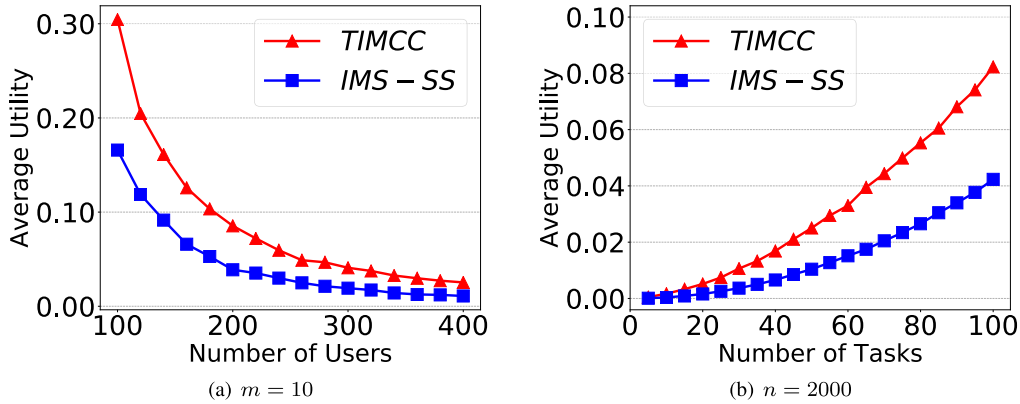


FIGURE 8. Average utility of TIMCC and IMS-SS.

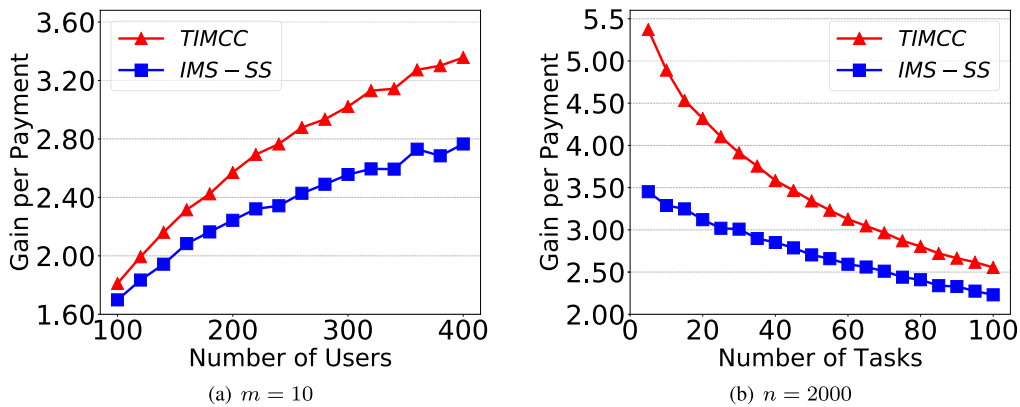


FIGURE 9. Platform gain pre payment of TIMCC and IMS-SS.

TIMCC determines the payment by the ratio of value to bid price, which will lead to a little higher payment compared to IMC-SS. Consider TIMCC provides a higher payment than IMC-SS, TIMCC may be easier to attract users in practice.

Fig. 9 shows the comparison of gain per payment in TIMCC and IMS-SS. We observe that the gain per payment increases with n , but decreases with m , which match our previous analysis. We also observe that the gain per payment of TIMCC is always higher than IMS-SS, it is natural because TIMCC selects winning bids according to the ratio of value to bid price, but IMS-SS only considers the bid price. In other words, TIMCC chooses the bids that can bring high gain per payment, but IMS-SS chooses the bids that are only cheap.

2) PROBABILITY OF FAILURE

Note that if there are not enough users, crowdsensing fails to work. In Fig. 10, we generate the bid set randomly. Let TIMCC and IMS-SS run on the bid set. We run the test for 1000 times and record the times that TIMCC and IMS-SS fail separately. We use frequency as an estimation of the *probability of failure*. In Fig. 10(a), we let n vary from 400 to 1400 with a step of 200. It is observed that when there are not enough users (eg. $n = 400$), both TIMCC and IMS-SS

fail many times. With the increment of the number of users n , the failing times of TIMCC and IMS-SS drop. However, IMS-SS fails more than TIMCC. In Fig. 10(b), we fix $|n| = 1000$, and m varies from 6 to 16 with a step of 2. While m grows, the platform needs more users, so both TIMCC and IMS-SS fail more times. In this case, IMS-SS fails more than TIMCC too. This is because TIMCC allows a bid to win more than once, making it perform much better, especially when there are not enough users.

VI. RELATED WORK

The design of incentive mechanisms for crowdsensing has attracted a lot of interests from the research community [9], [11]–[13], [23], [25]–[30]. We summarize incentive mechanisms for crowdsensing or crowdsourcing into two categories.

Incentive mechanisms based on auctions. In [23], the authors presented three models of crowdsensing, namely, Single-requester Multiple-bid model (SM-model), Multiple-requester Multiple-bid model (MM-model), and Single-requester Single-bid model (SS-model), where cooperation of providers are required. They designed an incentive mechanism for each of these models and proved that these incentive

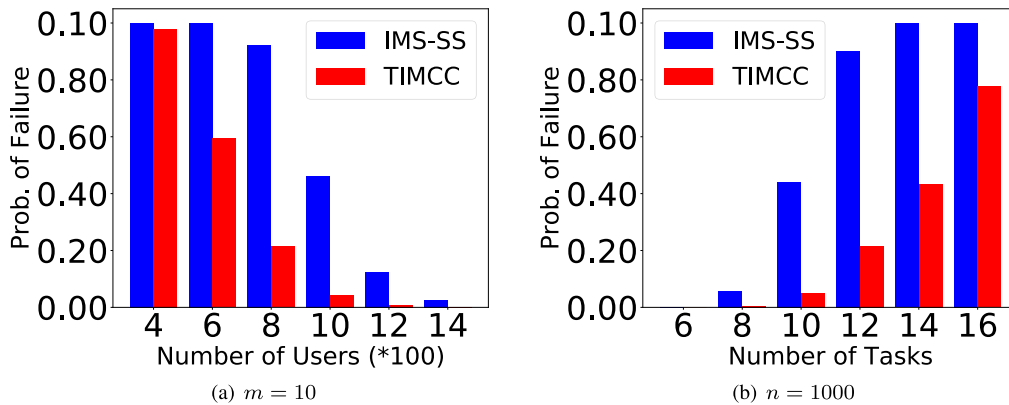


FIGURE 10. Probability of failure of TIMCC and IMS-SS.

mechanisms are individually rational, budget-balanced, computationally efficient, and truthful. As an extension of [23], Zhang *et al.* [12] presented an incentive mechanism TAFE (Truthful Auction with countermeasures against False-name Attacks). They also proved that TAFE is both truthful and robust against false-name attacks. Jaimes *et al.* [29] considered the locations of users, the coverage, and the budget constraint. They combined the Reverse Auction Dynamic Price with Recruitment (RADP-VPC-RC) mechanism [31] with the Greedy Budgeted Maximum Coverage to create a new Greedy Incentive Algorithm (GIA). Feng *et al.* [13] designed an algorithmic mechanism for inducing location-aware collaborative sensing in mobile crowdsourcing. They proved that optimally determining the winning bids with the location is *NP-hard* and proposed a near-optimal task allocation algorithm and a payment scheme that guarantees truthfulness. Singer and Mittal [27] proposed two online mechanisms to maximize the number of completed tasks while minimizing the total payment without exceeding the budget. The authors in [25] designed **RIT** (Robust Incentive Tree Mechanism for Crowdsensing) that motivates users for both participation and solicitation. They also proved that RIT is sybil-proof with probability at least H , for any given $H \in (0, 1)$. However, all those mechanisms regard tasks as one-shot tasks, it is not very suitable to apply those models to continuous crowdsensing, where tasks require continuous efforts. Our work is in the interest of filling this gap.

Incentive mechanisms without auctions. In [32], the authors proposed a non-monetary incentive mechanism to motivate smartphones to join the crowdsourcing system, where users are required to offer services to others if the user wants to receive services from others. Zhang *et al.* [33] integrated reputation into existing pricing schemes, and they used a rigorous repeated game framework to analyze users' current and future behavior, which achieved an improvement in performance of the noncooperative equilibria.

VII. FUTURE DIRECTION

In the future, we are particularly interested in further investigating the following directions.

Latency Control. Crowdsensing data may not be sensed very timely, where the reasons can be explained from different aspects. As mentioned in [34], one strategy for latency control is pricing, a higher price attracts more workers and can reduce the latency. However, a higher price naturally increases the financial burden of the platform at the same time. Another strategy for latency is a solicitation, *i.e.*, incentivizing users for both participation and solicitation. However, solicitation brings some challenges to the robustness of price manipulation.

Sybil Attack. As mentioned in [25], a sybil attack is that a user may generate multiple fake identities, and gain a higher utility without devoting extra efforts. The existence of sybil attack not only increases the burden of the platform, but also harms the interests of the honest users. The robustness against sybil attack is also very important to incentive mechanisms. However, it is beyond the scope of this paper and subjects to future researches.

VIII. CONCLUSION

In this paper, we considered continuous crowdsensing, where users are required to contribute continuous efforts. We defined a metric age of data to capture the freshness of data. We proposed a privacy-preserving incentive mechanism TIMCC for continuous crowdsensing that takes the age of data into consideration to reflect of the value of data, which further determines the rewards in the auction. We also proved that TIMCC satisfies individual rationality, computational efficiency, and truthfulness in theory, which is validated by extensive simulations.

REFERENCES

- [1] M. Chen, J. Zhou, G. Tao, J. Yang, and L. Hu, "Wearable affective robot," *IEEE Access*, vol. 6, pp. 64766–64776, 2018.
- [2] H. Hassan, J.-M. Martinez-Rubio, A. Perles, J.-V. Capella, C. Dominguez, and J. Albaladejo, "Smartphone-based industrial informatics projects and laboratories," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 557–566, Feb. 2013.
- [3] C. Wu, Z. Yang, and Y. Liu, "Smartphones based crowdsourcing for indoor localization," *IEEE Trans. Mobile Comput.*, vol. 14, no. 2, pp. 444–457, Feb. 2015.
- [4] K. Chen, C. Wang, Z. Yin, H. Jiang, and G. Tan, "Slide: Towards fast and accurate mobile fingerprinting for Wi-Fi indoor positioning systems," *IEEE Sensors J.*, vol. 18, no. 3, pp. 1213–1223, Feb. 2018.

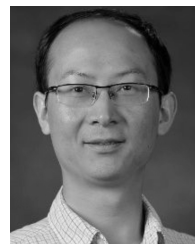
- [5] M. Stevens and E. D'Hondt, "Crowdsourcing of pollution data using smartphones," in *Proc. Workshop Ubiquitous Crowdsourcing*, 2010.
- [6] H. Jiang, C. Cai, X. Ma, Y. Yang, and J. Liu, "Smart home based on WiFi sensing: A survey," *IEEE Access*, vol. 6, pp. 13317–13325, 2018.
- [7] P. Zhou, Y. Zheng, and M. Li, "How long to wait?: Predicting bus arrival time with mobile phone based participatory sensing," *IEEE Trans. Mobile Comput.*, vol. 13, no. 6, pp. 1228–1241, Jun. 2014.
- [8] M. Chen, J. Yang, L. Hu, M. S. Hossain, and G. Muhammad, "Urban healthcare big data system based on crowdsourced and cloud-based air quality indicators," *IEEE Commun. Mag.*, vol. 56, no. 11, pp. 14–20, Nov. 2018.
- [9] X. Zhang, G. Xue, D. Yang, and R. Yu, "A sybil-proof and time-sensitive incentive tree mechanism for crowdsourcing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.
- [10] F. Restuccia, S. K. Das, and J. Payton, "Incentive mechanisms for participatory sensing: Survey and research challenges," *ACM Trans. Sensor Netw.*, vol. 12, no. 2, Apr. 2016, Art. no. 13.
- [11] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao, "Incentives for mobile crowd sensing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 54–67, 1st Quart., 2016.
- [12] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, "Countermeasures against false-name attacks on truthful incentive mechanisms for crowdsourcing," *IEEE J. Select. Areas Commun.*, vol. 35, no. 2, pp. 478–485, Feb. 2017.
- [13] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, "TRAC: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 1231–1239.
- [14] Y.-P. Hsu, E. Modiano, and L. Duan, "Age of information: Design and analysis of optimal scheduling algorithms," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 561–565.
- [15] A. M. Bedewy, Y. Sun, and N. B. Shroff, "Age-optimal information updates in multihop networks," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 576–580.
- [16] C. Kam, S. Kompella, and A. Ephremides, "Age of information under random updates," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2013, pp. 66–70.
- [17] N. Lu, B. Ji, and B. Li, "Age-based scheduling: Improving data freshness for wireless real-time traffic," in *Proc. 18th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2018.
- [18] M. Chen, Y. Hao, C.-F. Lai, D. Wu, Y. Li, and K. Hwang, "Opportunistic task scheduling over co-located clouds in mobile environment," *IEEE Trans. Serv. Comput.*, vol. 11, no. 3, pp. 549–561, May 2018.
- [19] C. Xu, X. Ma, R. Shea, H. Wang, and J. Liu, "Enhancing performance and energy efficiency for hybrid workloads in virtualized cloud environment," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2018.2837040.
- [20] V. Bindshaedler and R. Shokri, "Synthesizing plausible privacy-preserving location traces," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 546–563.
- [21] Q. He, G. Dan, and V. Fodor, "Minimizing age of correlated information for wireless camera networks," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2018, pp. 547–552.
- [22] M. Chen, Y. Hao, H. Gharavi, and V. C. Leung, "Cognitive information measurements: A new perspective," *Inf. Sci.*, vol. 505, pp. 487–497, Dec. 2019.
- [23] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, "Truthful incentive mechanisms for crowdsourcing," in *Proc. IEEE INFOCOM*, Apr./May 2015, pp. 2830–2838.
- [24] H. R. Varian, "Position auctions," *Int. J. Ind. Org.*, vol. 25, no. 6, pp. 1163–1178, Dec. 2007.
- [25] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, "Robust incentive tree design for mobile crowdsensing," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 458–468.
- [26] L. Gao, F. Hou, and J. Huang, "Providing long-term participation incentive in participatory sensing," in *Proc. IEEE INFOCOM*, Apr./May 2015, pp. 2803–2811.
- [27] Y. Singer and M. Mittal, "Pricing mechanisms for crowdsourcing markets," in *Proc. 22nd Int. Conf. World Wide Web (WWW)*, 2013.
- [28] Z. Feng, Y. Zhu, Q. Zhang, H. Zhu, J. Yu, J. Cao, and L. M. Ni, "Towards truthful mechanisms for mobile crowdsourcing with dynamic smartphones," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun./Jul. 2014, pp. 11–20.
- [29] L. G. Jaimes, I. Vergara-Laurens, and M. A. Labrador, "A location-based incentive mechanism for participatory sensing systems with budget constraints," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, Mar. 2012, pp. 103–108.
- [30] D. D'Alcantara and M. Vojnovic, "Crowdsourcing and all-pay auctions," in *Proc. 10th ACM Conf. Electron. Commerce (EC)*, 2009.
- [31] J.-S. Lee and B. Hoh, "Dynamic pricing incentive for participatory sensing," *Pervasive Mobile Comput.*, vol. 6, no. 6, pp. 693–708, Dec. 2010.
- [32] A. J. Mashhadi, S. B. Mokhtar, and L. Capra, "Fair content dissemination in participatory DTNs," *Ad Hoc Netw.*, vol. 10, no. 8, pp. 1633–1645, Nov. 2012.
- [33] Y. Zhang and M. Van Der Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2140–2148.
- [34] G. Li, J. Wang, Y. Zheng, and M. J. Franklin, "Crowdsourced data management: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2296–2319, Sep. 2016.



XIAOQIANG MA received the B.E. degree from the Huazhong University of Science and Technology, China, in 2010, and the M.Sc. and Ph.D. degrees from Simon Fraser University, Canada, in 2012 and 2015, respectively. His areas of interest are wireless networks, social networks, and cloud computing.



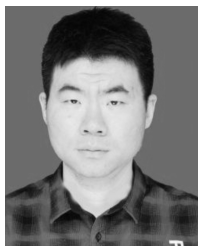
WEIWEI DENG received the B.E. degree in physics from the Huazhong University of Science and Technology, Wuhan, China, in 2012, where he is currently pursuing the M.S. degree with the School of Electronic Information and Communications. His current research interests include the Internet of Things, crowdsourcing, and machine learning.



FENG WANG received the bachelor's and master's degrees in computer science and technology from Tsinghua University, Beijing, China in 2002 and 2005, respectively, and the Ph.D. degree in computing science from Simon Fraser University, Burnaby, BC, Canada, in 2012. He is currently an Associate Professor with the Department of Computer and Information Science, The University of Mississippi, MS, USA. He is a Technical Committee Member of *Elsevier Computer Communications*. He also serves as a TPC member in various international conferences such as IEEE INFOCOM, ICPP, IEEE/ACM IWQoS, ACM Multimedia, IEEE ICC, IEEE GLOBECOM, and IEEE ICME. He was a recipient of the IEEE ICME Quality Reviewer Award, in 2011, and the ACM BuildSys Best Paper Award, in 2018. He served as a Program Vice Chair in International Conference on Internet of Vehicles (IOV) 2014, and as a TPC Co-Chair in IEEE CLOUDCOM 2017 for Internet of Things and Mobile on Cloud track.



MENGLAN HU received the B.E. degree in electronic and information engineering from the Huazhong University of Science and Technology, China, in 2007, and the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2012. He is currently an Associate Professor with the School of Electronic Information and Communications, Huazhong University of Science and Technology. His research interests include cloud computing, mobile computing, parallel and distributed systems, scheduling and resource management, and wireless networking.



FEI CHEN received the M.S. degree from the School of Electronics Engineering and Computer Science, Northeastern University, Shenyang, China, in 2009, and the Ph.D. degree from the School of Computing Science, Simon Fraser University, Burnaby, BC, Canada, in 2014. He is currently an Associate Professor in computing science technology with Qingdao University, China. His research interests include cloud computing, peer-to-peer networks, crowd-sensing systems, and multimedia communications.



MOHAMMAD MEHEDI HASSAN received the Ph.D. degree in computer engineering from Kyung Hee University, South Korea, in February 2011. He is currently an Associate Professor with the Information Systems Department, College of Computer and Information Sciences (CCIS), King Saud University (KSU), Riyadh, Saudi Arabia. He has published more than 130 research articles in the journals and conferences of international repute. He has also played role of the Guest Editor of several international ISI-indexed journals. His research areas of interest are cloud federation, multimedia cloud, sensor-cloud, the Internet of Things, Big data, mobile cloud, sensor networks, publish/subscribe systems, and recommender systems. He received the Best Journal Paper Award from IEEE SYSTEMS Journal in 2018. He also received the Best Paper Award from CloudComp Conference at China in 2014, and the Excellence in Research Award from CCIS, KSU, in 2015 and 2016, respectively.

...