Exploiting Data Reuse in Mobile Crowdsensing

Changkun Jiang*, Lin Gao†, Lingjie Duan‡, Jianwei Huang*
*Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong
†School of Electronic and Information Engineering, Harbin Institute of Technology (Shenzhen), China
‡Pillar of Engineering Systems and Design, Singapore University of Technology and Design, Singapore
Email: *{jc012, jwhuang}@ie.cuhk.edu.hk, †gaolin@hitsz.edu.cn, ‡lingjie_duan@sutd.edu.sg

***Abstract*—Mobile crowdsensing emerges as a promising sensing paradigm through leveraging the diverse embedded sensors in massive mobile devices. A key objective in mobile crowdsensing is to efficiently schedule mobile device users to perform multiple sensing tasks. Prior work mainly focused on the interactions between the task layer and the user layer, without considering the similarity of tasks' data requirements and the heterogeneity of users' sensing capabilities. In this work, we propose a three-layer data-centric crowdsensing model by introducing a new data layer between tasks and users, which allows us to effectively leverage both the task similarity and the user heterogeneity. We formulate a joint task selection and user scheduling problem on top of the data layer, aiming at maximizing the social welfare. This problem is difficult to solve due to the combinatorial nature as well as the two-sided private information of tasks and users. To address both issues, we propose a two-sided *randomized auction* mechanism, which is computationally efficient, individually rational, and incentive compatible in expectation. Simulations show that (i) the proposed randomized auction can achieve 90% of the maximum social welfare (benchmark), and (ii) the social welfare gain due to data reuse increases with the task similarity and reaches up to 1300% in our simulations.**

I. Introduction

The proliferation of hand-held mobile devices with rich embedded sensors has enabled a new sensing paradigm known as *Mobile CrowdSensing (MCS)* [1], where individual mobile users are involved to perform the sensing tasks by using their mobile devices. In a general multi-task MCS system (e.g., PRISM [2] and Medusa [3]), each sensing task is first initiated and announced by a task planner or owner via a web portal, and then assigned to a pool of mobile users (registered in the system), who will perform the sensing task accordingly (e.g., sensing the required data and sending the collected data to the system). While performing a sensing task, users consume their device resources such as battery energy and CPU time, hence incur certain costs. Therefore, users may not be willing to participate in the MCS system, unless they can receive satisfactory rewards to compensate their costs.

Many prior studies (e.g., [4]–[9]) have studied on the problem of incentivizing users to participate in the MCS system. These works focused on the interactions of tasks and users (e.g., the assignment of tasks among users through a proper matching), without considering the common data requirements across multiple tasks and the heterogeneous sensing capabilities of different users. In a practical system, however, there is a high likelihood that multiple tasks require some common data [1]. For example, the road traffic data at a particular time and location may be useful for Waze, Uber, and Google Traffic. Thus, it is likely to cause duplicated sensing and processing in a multi-task scenario, if multiple tasks are performed separately by the same user. Moreover, in practice, users may have different sensing capabilities due to factors such as locations and device types. For example, it is easier for a user to sense the data close to its current location. Thus, it is more flexible and efficient to schedule users on the data level than on the task level.

To complete multiple tasks more efficiently, it is critical to identify the common data requirements of these tasks and enable the reuse of sensory data across different tasks. Specifically, some practical MCS platforms (e.g., PRISM [2] and Medusa [3]) have allowed task developers to specify their data requirements in a high-level language. Then, they identify and reuse the common data across multiple tasks in order to reduce or avoid duplicated sensing and processing. There are several advantages to enable data reuse in the MCS system. First, data is digital goods and can be reused without additional cost. Second, tasks can share a large pool of mobile users collectively through the platform. Third, by reusing data across tasks, the overall system efficiency can be improved.

To this end, we propose a novel three-layer *data-centric* MCS model, consisting of a data layer, a task layer, and a user layer, which is different from the traditional two-layer *task-centric* model with the task layer and the user layer only [4]–[9]. Specifically, in our data-centric model, tasks and users are connected through the data layer, that is, each task is translated to a set of data items that it requires, and each user is associated with a set of data items that it can sense. Moreover, different tasks may require a common data item (hence can reuse the data item), and different users may be able to sense the same data item (hence compete for the sensing opportunity). Thus, it captures both the *task similarity* (data requirements) and the *user heterogeneity* (sensing capabilities). Fig. 1 illustrates the model with 6 tasks, 6 users, and 8 data items, where task 1 requires data items $\{1, 2\}$, and user 1 is able to sense data items $\{1, 2, 3\}$.

In such a data-centric model, the MCS platform collects the data requirements of tasks and the sensing capabilities of users, and then decides whether and how to complete these tasks by a proper set of users efficiently. Formally,

- *Which tasks can be completed?*
- *Which users will be scheduled for sensing which data?*

We focus on the optimal task selection and user scheduling that maximize the social welfare (defined as the difference

This work is supported by the General Research Fund (Project Number CUHK 14206315) established under the University Grant Committee of the Hong Kong Special Administrative Region, China.

978-1-5090-1328-9/16/$31.00 ©2016 IEEE

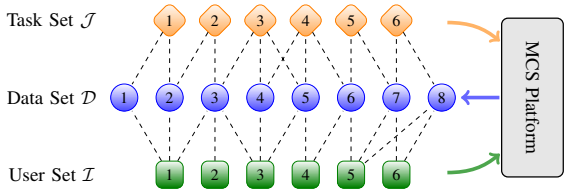

Fig. 1. Three-layer data-centric mobile crowdsensing model

between the total values of completed tasks and the total costs of scheduled users). Yet, solving the problem is challenging. First, it is NP-hard due to the combinatorial nature. Second, it requires the complete information regarding tasks' values and users' costs, which are often private information of task owners and users, respectively. Hence, a *truthful* incentive mechanism is necessary for eliciting such private information from task owners and users. However, some well-known truthful mechanisms (e.g., VCG [14]) are not suitable for our problem due to the high computational complexity.

To address the above issues, we resort to the *randomized auction* framework [10], with the MCS platform acting as the *auctioneer* and the participating tasks and users acting as the *bidders*. We propose a truthful randomized auction, consisting of (i) a randomized allocation rule, which picks up a feasible "allocation" randomly from a set of target solutions according to some probability distribution, and (ii) a payment rule, which assigns a payment for each bidder under any possible allocation. Randomized auctions have been widely adopted in wireless networking [11] and cloud computing [12]. Different from the auction models in [11], [12] (which are single-sided), our randomized auction is *two-sided*, in the sense that we need to decide both task selection (values) and user scheduling (costs) under mutual information asymmetry. Our main results and key contributions are summarized as follows.

- *Novel Data-Centric Crowdsensing Model:* To our best knowledge, this is the first work that analytically exploits data reuse across multiple tasks in MCS. We propose a novel three-layer data-centric model, which captures both the task similarity and the user heterogeneity.
- *Randomized Auction Design and Analysis:* To elicit the private information of both tasks and users, we propose a truthful two-sided randomized auction mechanism, which is computationally efficient, individually rational, and truthful in expectation.
- *Observations and Insights:* Simulations show that (i) our proposed randomized auction can achieve 90% of the maximum social welfare benchmark, and (ii) the social welfare gain due to data reuse increases with the task similarity and reaches up to 1300% in our simulations. Furthermore, with data reuse, the social welfare increases with the task similarity, as highly similar tasks can be potentially completed by a smaller set of users; while without data reuse, the social welfare decreases with the task similarity, due to the increased user competition.

The rest of the paper is organized as follows. In Section II, we present the system model. In Section III, we propose the randomized auction mechanism design. We present the simulations in Section IV, and conclude in Section V.

II. System Model

A. Network Model

We consider a general multi-task MCS model consisting of a set $\mathcal{J} = \{1, \cdots, J\}$ of tasks, a set $\mathcal{I} = \{1, \cdots, I\}$ of mobile users, and a set $\mathcal{D} = \{1, \cdots, K\}$ of target data items. Each data item $k \in \mathcal{D}$ is captured by a set of fine-grained parameters such as the data type, location, and time. Each task $j \in \mathcal{J}$ is associated with a set of data requirements $\mathcal{D}_j \subseteq \mathcal{D}$, and each user $i \in \mathcal{I}$ is able to sense a specific set $\mathcal{S}_i \subseteq \mathcal{D}$ of data items. As different tasks can reuse the same data items, there may exist multiple tasks j_1 and j_2 with overlapping data requirements, i.e., $\mathcal{D}_{j_1} \bigcap \mathcal{D}_{j_2} \neq \emptyset$. Fig. 1 illustrates such a three-layer data-centric MCS model.

The crowdsensing model operates in a time-slotted manner, where the whole time period is divided into multiple *time slots*, and each time slot can be an hour or a day, depending on the variations of tasks or users. Without loss of generality, we consider the operations in a particular time slot in this paper. At the beginning of the time slot, (i) each task owner registers its task on the platform, indicating the data requirements of the task and the potential value that he can achieve when the task is completed; and (ii) each user reports its information on the platform, indicating the sensing capability of the user (i.e., the set of data items that he can sense) and the potential cost of sensing any subset of data items within its capability. After collecting the reported information of all tasks and users, the platform decides the task selection (i.e., selecting a set of tasks to be completed) and the user scheduling (i.e., scheduling a set of users to sense the associated data items).

It is important to note that task owners or users may misreport their information to seek improvements of their individual benefits. We consider a simple scenario in this paper: (i) *users may misreport their sensing costs, but not their sensing capabilities*; and (ii) *task owners may misreport their values, but not their data requirements* (as the platform can check the correctness of these information afterwards).

B. Task Model

Each task $j \in \mathcal{J}$ is associated with a set of data requirements $\mathcal{D}_j \subseteq \mathcal{D}$ in the time slot that we focus on, and a task value $v_j > 0$ when it is completed. The task value v_j is the *private information* of task j, and cannot be observed by the platform, users, or other tasks. We assume that a task j is completed if and only if each of its required data items in $\mathcal{D}_j$ has been sensed by at least one user. Let $z_j \in \{0, 1\}$ denote whether a task $j \in \mathcal{J}$ is completed, and $y_k \in \{0, 1\}$ denote whether a data item $k \in \mathcal{D}$ is sensed by at least one user. Then, for each task $j \in \mathcal{J}$, we have the following constraint:

$$z_j \leq y_k, \quad \forall k \in \mathcal{D}_j. \tag{1}$$

Given a feasible task selection $\boldsymbol{z} = (z_j, j \in \mathcal{J})$, the total achieved value (of all completed tasks) is:

$$V(\boldsymbol{z}) = \textstyle\sum_{j \in \mathcal{J}} v_j \cdot z_j. \tag{2}$$

C. User Model

Each user $i \in \mathcal{I}$ is associated with a sensing capability in the time slot that we focus on, i.e., a set $\mathcal{S}_i$ of data items that it can sense. The platform can schedule user i to sense a subset $\mathcal{S} \subseteq \mathcal{S}_i$ of data items within its sensing capability, associated with a sensing cost $c_i(\mathcal{S})$. Let $x_i(\mathcal{S}) \in \{0,1\}$ denote whether a user i is scheduled to sense a data set $\mathcal{S} \subseteq \mathcal{S}_i$. When $\mathcal{S} = \emptyset$, then $x_i(\emptyset) = 1$ denotes that user i is not scheduled to sense any data set, hence has a zero sensing cost, i.e., $c_i(\emptyset) = 0$.

We assume that a user can only be scheduled to sense one data set within its capability. That is, for each user $i \in \mathcal{I}$, we have the following user scheduling constraint:

$$\textstyle\sum_{\mathcal{S} \subseteq \mathcal{S}_i} x_i(\mathcal{S}) = 1. \tag{3}$$

Otherwise, if a user is scheduled to sense multiple data sets, say $\mathcal{S}_i^1$ and $\mathcal{S}_i^2$, we can always equivalently say that it is scheduled to sense the data set $\mathcal{S}_i^1 \bigcup \mathcal{S}_i^2$. Let $\boldsymbol{x}_i \triangleq (x_i(\mathcal{S}),\ \mathcal{S} \subseteq \mathcal{S}_i)$ denote the scheduling vector of user i. Then, given a feasible user scheduling $\boldsymbol{x} \triangleq (\boldsymbol{x}_i, i \in \mathcal{I})$, the total incurred sensing cost (of all scheduled users) is:

$$C(\boldsymbol{x}) = \textstyle\sum_{i \in \mathcal{I}} \sum_{\mathcal{S} \subseteq \mathcal{S}_i} c_i(\mathcal{S}) \cdot x_i(\mathcal{S}). \tag{4}$$

Let $y_{ki} \in \{0,1\}$ denote whether a data item k is sensed by a user i, that is, $y_{ki} = \sum_{\mathcal{S} \subseteq \mathcal{S}_i : k \in \mathcal{S}} x_i(\mathcal{S})$. Recall that $y_k \in \{0,1\}$ denotes whether a data item $k \in \mathcal{D}$ is sensed by at least one user. Then, for each data item $k \in \mathcal{D}$, we have the following constraint:

$$y_k \leq \textstyle\sum_{i \in \mathcal{I}} y_{ki}. \tag{5}$$

Moreover, we denote $\boldsymbol{c}_i \triangleq (c_i(\mathcal{S}),\ \mathcal{S} \subseteq \mathcal{S}_i)$ as the sensing cost vector of user i for all possible subsets of data items that it can sense. In practice, the sensing cost vector $\boldsymbol{c}_i$ is the *private information* of user i, and cannot be observed by the platform, tasks, or other users.[1] This is one of the key challenges for optimizing a crowdsensing system with data reuse.

D. Problem Formulation

The social welfare $W(\boldsymbol{x}, \boldsymbol{z})$ is defined as the difference between the total value $V(\boldsymbol{z})$ of all completed tasks and the total sensing cost $C(\boldsymbol{x})$ of all scheduled users, i.e.,

$$W(\boldsymbol{x}, \boldsymbol{z}) = V(\boldsymbol{z}) - C(\boldsymbol{x}). \tag{6}$$

The objective of the platform is to decide the best task selection $\boldsymbol{z}$ and user scheduling $\boldsymbol{x}$ that maximize the social welfare $W(\boldsymbol{x}, \boldsymbol{z})$. Formally, we can formulate such a joint task selection and user scheduling problem (P1) as follows.

$$\begin{aligned}
\text{P1:} \quad & \max_{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}} \quad V(\boldsymbol{z}) - C(\boldsymbol{x}) \\
& \text{s.t.} \quad (1)\text{–}(5), \quad \forall i \in \mathcal{I}, j \in \mathcal{J}, k \in \mathcal{D}; \\
& \text{var.} \quad x_i(\mathcal{S}) \in \{0,1\}, \quad \forall \mathcal{S} \in \mathcal{S}_i, i \in \mathcal{I}; \\
& \qquad\quad z_j \in \{0,1\}, \quad \forall j \in \mathcal{J}; \\
& \qquad\quad y_k \in \{0,1\}, \quad \forall k \in \mathcal{D}.
\end{aligned}$$

Here $\boldsymbol{y} \triangleq (y_k, k \in \mathcal{D})$ is an intermediate variable denoting whether each data item is sensed (by at least one user), which bridges the task selection and the user scheduling. It is easy to see that Problem P1 is a binary integer linear programming problem. Let $\{\boldsymbol{x}^o, \boldsymbol{y}^o, \boldsymbol{z}^o\}$ denote the optimal solution to P1. For clarity, we will also write $\{\boldsymbol{x}^o, \boldsymbol{y}^o, \boldsymbol{z}^o\}$ as $\{\boldsymbol{x}^o(\boldsymbol{c}, \boldsymbol{v}), \boldsymbol{y}^o(\boldsymbol{c}, \boldsymbol{v}), \boldsymbol{z}^o(\boldsymbol{c}, \boldsymbol{v})\}$, as they are functions of $\boldsymbol{c} \triangleq (\boldsymbol{c}_i, i \in \mathcal{I})$ and $\boldsymbol{v} \triangleq (v_j, j \in \mathcal{J})$.

However, solving Problem P1 is challenging. First, it is NP-hard (as the special case of P1 can be reduced to the NP-hard set cover problem), and hence it is necessary to design a low-complexity approximate algorithm to find an approximate solution. Second, solving Problem P1 requires the complete information including the data requirements and values of all tasks as well as the sensing capabilities and costs of all users. However, as we have mentioned earlier, users' sensing costs and tasks' values are their private information, and cannot be observed by the platform. Thus, we need to design a truthful incentive mechanism to elicit such private information.

III. Two-sided Auction-based Incentive Mechanism

In this section, we propose a two-sided auction-based incentive mechanism framework for solving Problem P1. First, within this framework, we propose a two-sided VCG auction mechanism (as benchmark) for solving Problem P1 exactly, without considering the complexity issue. Then, building upon the idea of the two-sided VCG auction, we further propose a low-complexity truthful randomized auction mechanism for solving Problem P1 approximately in polynomial time.

A. Two-sided Auction-based Mechanism Framework

To solve Problem P1 with two-sided private information, we propose a two-sided auction-based incentive mechanism, where the platform acts as an *auctioneer*, purchasing data from mobile users (*bidders* on one side) and selling data to tasks (*bidders* on the other side). In this auction framework, the platform first announces an *allocation rule* (for task selection and user scheduling) and a *payment rule* (for payments to the scheduled users and prices charged to the selected tasks). Then, each task submits a bid (indicating its value) and each user submits a bid (indicating its sensing cost vector) to the platform, which can be different from the true task value or user sensing cost vector, depending on whether the auction is truthful. Finally, the platform computes the allocations and payments, based on the reported bids of all tasks and users, together with other public information. In this work, we are interested in designing the *truthful* auction, where tasks and users submit their private information truthfully.

Next we provide the key notations. Let u_j denote the reported value (bid) of task j, and $\boldsymbol{b}_i \triangleq (b_i(\mathcal{S}),\ \mathcal{S} \subseteq \mathcal{S}_i)$ denote the reported sensing cost vector (bid) of user i, where $b_i(\mathcal{S})$ is the reported sensing cost for a data set $\mathcal{S} \subseteq \mathcal{S}_i$. Let $\boldsymbol{u} \triangleq (u_j, j \in \mathcal{J})$ denote the bids of all tasks and $\boldsymbol{b} \triangleq (\boldsymbol{b}_i, i \in \mathcal{I})$ denote the bids of all users. Obviously, we will have $\boldsymbol{b} = \boldsymbol{c}$ and $\boldsymbol{u} = \boldsymbol{v}$ at the equilibrium if the auction is truthful. With a little abuse of notation, we denote $\{\boldsymbol{x}(\cdot), \boldsymbol{z}(\cdot)\}$ as the allocation rule, where $\boldsymbol{x}(\cdot) \triangleq (\boldsymbol{x}_i(\cdot), i \in \mathcal{I})$

[1] As assumed previously, except the task values and the user sensing costs, all the other information (i.e., $\mathcal{S}_i, \mathcal{D}_j$) are *public information* to the platform.

is the user scheduling vector and $\boldsymbol{z}(\cdot) \triangleq (z_j(\cdot), j \in \mathcal{J})$ is the task selection vector. We further denote $\{\boldsymbol{p}(\cdot), \boldsymbol{q}(\cdot)\}$ as the payment rule, where $\boldsymbol{p}(\cdot) \triangleq (p_i(\cdot), i \in \mathcal{I})$ is the user payment vector, and $\boldsymbol{q}(\cdot) \triangleq (q_j(\cdot), j \in \mathcal{J})$ is the task charge vector. Note that $\boldsymbol{x}(\cdot)$, $\boldsymbol{z}(\cdot)$, $\boldsymbol{p}(\cdot)$, and $\boldsymbol{q}(\cdot)$ can also be written as $\boldsymbol{x}(\boldsymbol{b}, \boldsymbol{u})$, $\boldsymbol{z}(\boldsymbol{b}, \boldsymbol{u})$, $\boldsymbol{p}(\boldsymbol{b}, \boldsymbol{u})$, and $\boldsymbol{q}(\boldsymbol{b}, \boldsymbol{u})$, as they are all functions of $\boldsymbol{b}$ and $\boldsymbol{u}$. For convenience, we write such an auction mechanism as $\Omega \triangleq \{\boldsymbol{x}(\cdot), \boldsymbol{z}(\cdot); \boldsymbol{p}(\cdot), \boldsymbol{q}(\cdot)\}$ or $\Omega \triangleq \{\boldsymbol{x}(\boldsymbol{b}, \boldsymbol{u}), \boldsymbol{z}(\boldsymbol{b}, \boldsymbol{u}); \boldsymbol{p}(\boldsymbol{b}, \boldsymbol{u}), \boldsymbol{q}(\boldsymbol{b}, \boldsymbol{u})\}$.

B. Two-sided VCG Auction Mechanism (Benchmark)

We first propose a two-sided VCG auction mechanism (as benchmark) based on the idea of VCG auction [14].

We denote the two-sided VCG auction as $\Omega^o = \{\boldsymbol{x}(\cdot), \boldsymbol{z}(\cdot); \boldsymbol{p}(\cdot), \boldsymbol{q}(\cdot)\}$. Formally, its allocation rule is given by:

$$\boldsymbol{x}(\boldsymbol{b}, \boldsymbol{u}) = \boldsymbol{x}^o(\boldsymbol{b}, \boldsymbol{u}) \quad \text{and} \quad \boldsymbol{z}(\boldsymbol{b}, \boldsymbol{u}) = \boldsymbol{z}^o(\boldsymbol{b}, \boldsymbol{u}),$$

where $\{\boldsymbol{x}^o(\boldsymbol{b}, \boldsymbol{u}), \boldsymbol{z}^o(\boldsymbol{b}, \boldsymbol{u})\}$ is the optimal solution to Problem P1 by replacing $\boldsymbol{c}$ with $\boldsymbol{b}$ and $\boldsymbol{v}$ with $\boldsymbol{u}$. Moreover, its payment rule is given by:

$$\boldsymbol{p}(\boldsymbol{b}, \boldsymbol{u}) = \boldsymbol{p}^o(\boldsymbol{b}, \boldsymbol{u}) \triangleq (p_i^o(\boldsymbol{b}, \boldsymbol{u}))_{i \in \mathcal{I}},$$
$$\boldsymbol{q}(\boldsymbol{b}, \boldsymbol{u}) = \boldsymbol{q}^o(\boldsymbol{b}, \boldsymbol{u}) \triangleq (q_j^o(\boldsymbol{b}, \boldsymbol{u}))_{j \in \mathcal{J}},$$

where

$$p_i^o(\boldsymbol{b}, \boldsymbol{u}) \triangleq \sum_{j \in \mathcal{J}} u_j z_j^o(\boldsymbol{b}, \boldsymbol{u}) - \sum_{n \in \mathcal{I} \setminus \{i\}} \sum_{\mathcal{S} \subseteq \mathcal{S}_n} b_n(\mathcal{S}) x_n^o(\mathcal{S}) - W_{-i}^o,$$
$$q_j^o(\boldsymbol{b}, \boldsymbol{u}) \triangleq W_{-j}^o - \sum_{j \in \mathcal{J} \setminus \{j\}} u_j z_j^o(\boldsymbol{b}, \boldsymbol{u}) + \sum_{n \in \mathcal{I}} \sum_{\mathcal{S} \subseteq \mathcal{S}_n} b_n(\mathcal{S}) x_n^o(\mathcal{S}),$$

and W_{-i}^o is the maximum social welfare (defined on bids $\boldsymbol{b}, \boldsymbol{u}$) excluding user i's bid.[2]

We can prove that the two-sided VCG auction Ω^o is truthful and efficient (i.e., maximizes the social welfare). However, computing the two-sided VCG auction outcome needs to solve the NP-hard Problem P1, which is computationally expensive. To this end, we will propose a low-complexity truthful auction mechanism in the next subsection.

C. Truthful Randomized Auction Mechanism

Now we propose a low-complexity truthful randomized auction mechanism for solving Problem P1 in polynomial time. We start from the linear programming relaxation of Problem P1, obtaining an associated linear programming Problem P2 *in the fractional domain*, from which we further derive the fractional VCG auction (which may not be implementable in practice). Then, through proper decompositions, we transform the fractional VCG auction to the two-sided randomized auction (which is implementable).

1) ***Linear Programming Relaxation****:* We first relax Problem P1 to the fractional domain (i.e., every binary variable in the set $\{0, 1\}$ is relaxed to the interval $[0, 1]$), and denote the associated linear programming problem as Problem P2. Note that Problem P2 can be solved to its optimality in polynomial time as it is a standard linear programming [15]. We refer to the solution to Problem P2 as the *fractional optimal solution*, denoted by $\{\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{z}^*\}$ or $\{\boldsymbol{x}^*(\boldsymbol{c}, \boldsymbol{v}), \boldsymbol{y}^*(\boldsymbol{c}, \boldsymbol{v}), \boldsymbol{z}^*(\boldsymbol{c}, \boldsymbol{v})\}$. It is notable that the solution to Problem P2 provides an upper-bound for that to Problem P1, and the gap between them is called the *integrality gap* [15]. Intuitively, a fractional solution can be viewed as *the fraction of the time* that users are scheduled or tasks are selected.

Next, we propose the fractional VCG auction Ω^*, where the allocation rule aims to maximize the social welfare (defined on user bids $\boldsymbol{b}$ and task bids $\boldsymbol{u}$) *in the fractional domain*, and the payment rule aims to pay each scheduled user its social benefit and charge each selected task its social damage. The detailed mechanism is similar to Ω^o, except that we replace the integer solution $\{\boldsymbol{x}^o(\boldsymbol{b}, \boldsymbol{u}), \boldsymbol{z}^o(\boldsymbol{b}, \boldsymbol{u})\}$ by the fractional optimal solution $\{\boldsymbol{x}^*(\boldsymbol{b}, \boldsymbol{u}), \boldsymbol{z}^*(\boldsymbol{b}, \boldsymbol{u})\}$, and solving Problem P2 rather than Problem P1 when deciding the payments.

Proposition 1. *The fractional VCG auction Ω^* is individually rational, incentive compatible (truthful), and maximizes the social welfare in the fractional domain.*

Note that the optimal solution to Problem P2 (or the outcome of Ω^*) may *not* be feasible to Problem P1. This implies that *the fractional VCG auction Ω^* may not be implementable*. Next, we will transform Ω^* to the randomized auction, which always generates a feasible solution to Problem P1 randomly according to certain probability.

2) ***Randomized Mechanism Basics****:* We first introduce the key concepts in randomized mechanisms [10].

Recall that a two-sided *deterministic* mechanism $\Omega = \{\boldsymbol{x}(\cdot), \boldsymbol{z}(\cdot); \boldsymbol{p}(\cdot), \boldsymbol{q}(\cdot)\}$ consists of a deterministic allocation rule $\{\boldsymbol{x}(\cdot), \boldsymbol{z}(\cdot)\}$ and a payment rule $\{\boldsymbol{p}(\cdot), \boldsymbol{q}(\cdot)\}$, and returns a deterministic outcome $\{\boldsymbol{x}(\boldsymbol{b}, \boldsymbol{u}), \boldsymbol{z}(\boldsymbol{b}, \boldsymbol{u}); \boldsymbol{p}(\boldsymbol{b}, \boldsymbol{u}), \boldsymbol{q}(\boldsymbol{b}, \boldsymbol{u})\}$ given any bids $\boldsymbol{b}$ and $\boldsymbol{u}$. Note that both Ω^o and Ω^* mentioned above are deterministic mechanisms.

A mechanism $\widetilde{\Omega} \triangleq \{\widetilde{\boldsymbol{x}}(\cdot), \widetilde{\boldsymbol{z}}(\cdot); \widetilde{\boldsymbol{p}}(\cdot), \widetilde{\boldsymbol{q}}(\cdot)\}$ can also be randomized, in which it will flip coins to determine the allocation and payment, instead of giving out a deterministic allocation and payment. In other words, given any bids $\boldsymbol{b}$ and $\boldsymbol{u}$, the outcomes $\widetilde{x}_i(\boldsymbol{b}, \boldsymbol{u})$, $\widetilde{z}_j(\boldsymbol{b}, \boldsymbol{u})$, $\widetilde{p}_i(\boldsymbol{b}, \boldsymbol{u})$ and $\widetilde{q}_j(\boldsymbol{b}, \boldsymbol{u})$ are all random variables. Moreover, each task's utility (i.e., value minus charge) and each user's utility (i.e., payment minus sensing cost) are also random variables. Intuitively, such a randomized mechanism can be viewed as a set of randomizations over the deterministic mechanism. For randomized mechanisms, the concept of truthfulness is defined in the expected sense. That is, if a randomized mechanism $\widetilde{\Omega}$ is truthful in expectation, then the *expected* utilities of each user and each task are maximized when reporting truthfully.

3) ***Randomized Mechanism Design Criterion****:* We now provide the design criterion of a randomized mechanism.

We first introduce an (α, β)-scaled fractional mechanism for the (deterministic) mechanism $\Omega = \{\boldsymbol{x}(\cdot), \boldsymbol{z}(\cdot); \boldsymbol{p}(\cdot), \boldsymbol{q}(\cdot)\}$, inspired by the α-scaled fractional mechanism defined in [10], [11]. The key difference between the α-scaled fractional mechanism in [10], [11] and the (α, β)-scaled fractional mechanism is that the former one considers only the scaling of one side, while our mechanism considers the scaling of both sides.

[2]Specifically, W_{-i}^o is the maximizer of Problem P1, by replacing $\boldsymbol{c}$ with $\boldsymbol{b}$ and $\boldsymbol{v}$ with $\boldsymbol{u}$, and excluding user i's bid $\boldsymbol{b}_i$ before solving Problem P1.

Definition 1 (Scaled Fractional Mechanism)**.** *An (α, β)-scaled fractional mechanism of $\Omega = \{\boldsymbol{x}(\cdot), \boldsymbol{z}(\cdot); \boldsymbol{p}(\cdot), \boldsymbol{q}(\cdot)\}$, denoted by $\Omega_{(\alpha,\beta)} = \{\boldsymbol{x}_\alpha(\cdot), \boldsymbol{z}_\beta(\cdot); \boldsymbol{p}_\alpha(\cdot), \boldsymbol{q}_\beta(\cdot)\}$, is defined as:*

$$\boldsymbol{x}_\alpha(\cdot) = \alpha \cdot \boldsymbol{x}(\cdot), \quad \boldsymbol{p}_\alpha(\cdot) = \alpha \cdot \boldsymbol{p}(\cdot), \tag{7}$$

$$\boldsymbol{z}_\beta(\cdot) = \beta \cdot \boldsymbol{z}(\cdot), \quad \boldsymbol{q}_\beta(\cdot) = \beta \cdot \boldsymbol{q}(\cdot), \tag{8}$$

where $\alpha, \beta > 0$ are the scaling factors with $0 \preceq \alpha \cdot \boldsymbol{x}(\cdot) \preceq 1$ and $0 \preceq \beta \cdot \boldsymbol{z}(\cdot) \preceq 1$, respectively.

Intuitively, in an (α, β)-scaled fractional mechanism, the incurred cost and payment of each user are scaled with α, and the achieved value and charge of each task are scaled with β, compared with those in the original mechanism Ω. Thus,

Proposition 2. *If a mechanism Ω is truthful, then its (α, β)-scaled fractional mechanism $\Omega_{(\alpha,\beta)}$ is also truthful.*

Based on the above, we propose the following truthful randomized mechanism design criterion:

- *Design a randomized mechanism $\widetilde{\Omega}$ that generates the equivalent outcome (in terms of the task selection, user scheduling, and payment) as an (α, β)-scaled fractional mechanism $\Omega^*_{(\alpha,\beta)}$ of the fractional VCG auction Ω^*.*

As Ω^* is truthful, we can obtain the truthfulness of its (α, β)-scaled fractional mechanism $\Omega^*_{(\alpha,\beta)}$ by Proposition 2. Moreover, as $\widetilde{\Omega}$ generates the same task selection, user scheduling, and payment as $\Omega^*_{(\alpha,\beta)}$, we can further obtain the truthfulness (in expectation) of the randomized mechanism $\widetilde{\Omega}$.

4) ***Truthful Randomized Mechanism****:* Now we design the truthful randomized mechanism.

For convenience, we express a randomized mechanism $\widetilde{\Omega} = \{\widetilde{\boldsymbol{x}}(\cdot), \widetilde{\boldsymbol{z}}(\cdot); \widetilde{\boldsymbol{p}}(\cdot), \widetilde{\boldsymbol{q}}(\cdot)\}$ as a set of allocation probabilities $\boldsymbol{\lambda} = (\lambda^l)_{l \in \mathcal{A}}$ and a set of payment rules $\{\boldsymbol{p}^l(\cdot), \boldsymbol{q}^l(\cdot)\}_{l \in \mathcal{A}}$ under all possible allocations, where $\mathcal{A}$ is the set of all feasible *integer* allocations (regarding $\boldsymbol{x}$ and $\boldsymbol{z}$) and $\lambda^l \geq 0$ is the probability of picking up a particular allocation $\{\boldsymbol{x}^l, \boldsymbol{z}^l\}$ and an associated payment $\{\boldsymbol{p}^l, \boldsymbol{q}^l\}$. Then, designing a randomized mechanism $\widetilde{\Omega}$ is equivalent to finding a set of allocation probabilities $\boldsymbol{\lambda} = (\lambda^l)_{l \in \mathcal{A}}$ and a set of payment rules $\{\boldsymbol{p}^l(\cdot), \boldsymbol{q}^l(\cdot)\}_{l \in \mathcal{A}}$.

Next, we propose the randomized auction $\widetilde{\Omega}$, which aims to maximize the two-sided scaled social welfare subject to the exact decomposition of the fractional optimal solution into the weighted sum of integer solutions. Due to the two-sided social welfare maximization, $\widetilde{\Omega}$ nontrivially extends those with one-sided utility maximization or cost minimization in [10], [11].

Mechanism (Randomized Auction Mechanism – $\widetilde{\Omega}$)**.**

- *Allocation Rule $\widetilde{\boldsymbol{\lambda}} = (\lambda^l)_{l \in \mathcal{A}}$:*

$$\begin{aligned} \widetilde{\boldsymbol{\lambda}} = \arg \max_{\boldsymbol{\lambda}, 0 < \alpha, \beta \leq 1} \quad & \beta \cdot V^* - \alpha \cdot C^* \\ \text{s.t., } & \textstyle\sum_{l \in \mathcal{A}} \lambda^l \cdot \boldsymbol{x}_i^l = \alpha \cdot \boldsymbol{x}_i^*(\boldsymbol{b}, \boldsymbol{u}), \quad \forall i \in \mathcal{I}, \\ & \textstyle\sum_{l \in \mathcal{A}} \lambda^l \cdot z_j^l = \beta \cdot z_j^*(\boldsymbol{b}, \boldsymbol{u}), \quad \forall j \in \mathcal{J}, \end{aligned}$$

  where V^* and C^* are the optimal total task values and user costs w.r.t. $\boldsymbol{z}^*(\boldsymbol{b}, \boldsymbol{u})$ and $\boldsymbol{x}^*(\boldsymbol{b}, \boldsymbol{u})$, respectively.

- *Payment Rule $\{\boldsymbol{p}^l(\boldsymbol{b}, \boldsymbol{u}), \boldsymbol{q}^l(\boldsymbol{b}, \boldsymbol{u})\}_{l \in \mathcal{A}}$:*

$$p_i^l(\boldsymbol{b}, \boldsymbol{u}) = \alpha \cdot p_i^*(\boldsymbol{b}, \boldsymbol{u}) \cdot \frac{C_i(\boldsymbol{x}_i^l)}{\sum_{l' \in \mathcal{A}} \lambda_{l'} \cdot C_i(\boldsymbol{x}_i^{l'})}, \forall i \in \mathcal{I},$$

$$q_j^l(\boldsymbol{b}, \boldsymbol{u}) = \beta \cdot q_j^*(\boldsymbol{b}, \boldsymbol{u}) \cdot \frac{V_j(z_j^l)}{\sum_{l' \in \mathcal{A}} \lambda_{l'} \cdot V_j(z_j^{l'})}, \forall j \in \mathcal{J},$$

  where $C_i(\boldsymbol{x}_i^l)$ is user i's cost under the allocation $\boldsymbol{x}_i^l$, and $V_j(z_j^l)$ is task j's value under the allocation z_j^l.

Obviously, both the expected payment and sensing cost of each user *and* the expected charge and value of each task in $\widetilde{\Omega}$ are equivalent to those in $\Omega^*_{(\alpha,\beta)}$, which implies that $\widetilde{\Omega}$ is truthful (incentive compatible) *in expectation*, in the sense that each user and task can maximize their expected utilities when reporting truthfully. Formally,

Proposition 3. *The randomized mechanisms $\widetilde{\Omega}$ is incentive compatible (truthful) in expectation.*

We can further check that under the mechanism $\widetilde{\Omega}$, each user and task can always achieve a non-negative utility *under any possible realization of allocations*. This implies that $\widetilde{\Omega}$ is individually rational *in the strict sense*. Formally,

Proposition 4. *The randomized mechanisms $\widetilde{\Omega}$ is individually rational in the strict sense.*

Furthermore, we can see that in $\widetilde{\Omega}$, each user's sensing cost equals α^* times the sensing cost incurred in Ω^*, while each task's value equals β^* times the value achieved in Ω^* (where α^*, β^* are the optimal solutions to the allocation problem in $\widetilde{\Omega}$). Hence, the efficiency of $\widetilde{\Omega}$ is guaranteed in this sense.

Proposition 5 (Efficiency of $\widetilde{\Omega}$)**.** *The mechanism $\widetilde{\Omega}$ guarantees to achieve a β^*-fraction of the total task values in Ω^* with an α^*-fraction of the total sensing costs in Ω^*.*

IV. Simulation Results

Now we provide simulation results to evaluate the performance of our proposed randomized auction mechanism.

A. Simulation Setup

In the simulations, we fix the number of tasks to $J = 50$ and the number of data items to $K = 30$, while varying the number of users from $I = 10$ to 100 with an increment of 10. Each data item is location-based, and distributed in an area of $1000\text{m} \times 1000\text{m}$ randomly and uniformly. Each user randomly moves to a particular location in a time slot, and can sense all the data items within a distance of 100m to its location. The unit cost ρ_c of each user for sensing one data item is chosen randomly from $[1, 5]$, hence the cost for sensing a set $\mathcal{S}$ of data items is $\rho_c \cdot |\mathcal{S}|$. The unit value ρ_v of each task for one data item is also chosen randomly from $[1, 5]$, hence the value of a task requiring a set $\mathcal{S}$ of data items is $\rho_v \cdot |\mathcal{S}|$.

We characterize task similarity (in terms of data requirements) in the following way. We define the popularity of a data item as the probability that a task requires this particular data item, and denote p_k as the k-th highest popularity of all data items. As demonstrated in [13], the popularity of data, i.e., $\{p_k, k \in \mathcal{D}\}$, follows a Zipf distribution with the p.m.f.

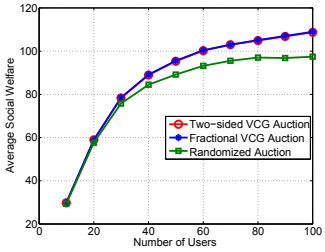

Fig. 2. Social welfare vs user number

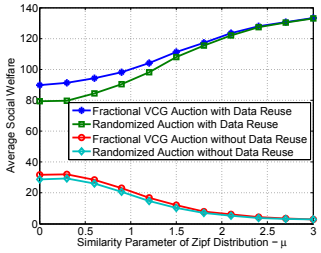

Fig. 3. Social welfare vs task similarity

$p_k = (\frac{1}{k})^\mu / \sum_{t=1}^{K} (\frac{1}{t})^\mu, \forall k \in \mathcal{D}$, where $\mu \geq 0$ is the parameter of Zipf distribution. Obviously, with a larger μ, tasks are more likely to require a small set of high popularity data items (hence with a higher task similarity); while with a smaller μ, tasks are more likely to require data items equally (hence with a lower task similarity). In our simulations, we vary μ from 0 to 3 with an increment of 0.3.

In each simulation, we select a particular I and μ, and compute the result by averaging over 1000 randomly generated systems (e.g., tasks' data requirements and values, users' sensing capabilities and costs).

B. Social Welfare Gap

We first illustrate the average social welfare achieved in the randomized auctions, the two-sided VCG, and the fractional VCG auction in Fig. 2. This can help us to understand the performance gap of our proposed randomized auction to the maximum social welfare (achieved in the two-sided VCG auction) or the fractional maximum social welfare (achieved in the fractional VCG auction). From Fig. 2, we can see that the gap between maximum and fractional maximum social welfare is negligible. Moreover, the gap of our proposed randomized auction to the maximum social welfare benchmark increase with the number of users, and the maximal gap in our simulations (under 100 users) is less than 10%.

C. Performance Gain of Data Reuse

We now evaluate the performance gain due to data reuse, by comparing the social welfare with and without data reuse. Note that without data reuse, a data item provided by a user can only be used by one task. Hence, if multiple tasks require a data, the data has to be sensed multiple times by one or multiple users. This implies the following additional constraint: for each data $k \in \mathcal{D}$, if it is required by M (selected) tasks, then it has to be sensed by at least M times, i.e.,

$$\sum_{j \in \mathcal{J}} \mathbf{1}_{(k \in \mathcal{D}_j)} \cdot z_j \leq \sum_{i \in \mathcal{I}} \mathbf{1}_{(k \in \mathcal{S})} \cdot x_i(\mathcal{S}), \quad \forall k \in \mathcal{D},$$

where the indicator $\mathbf{1}_{(x)} = 1$ if x is true, and 0 otherwise.

We next show the impact of task similarity on the performance gain. Recall that the parameter μ of Zipf reflects the task similarity: *a larger μ implies a higher task similarity*. Fig. 3 illustrates the social welfare with and without data reuse, under different values of μ, where the user number is fixed at $I = 60$. From Fig. 3, we can see that the achieved social welfare increases with μ with data reuse, while decreases with μ without data reuse. The reason is as follows. With a higher task similarity μ, most of the tasks' data requirements will concentrate on a smaller set of high popularity data. Hence, with data reuse, a smaller set of users (covering the high popularity data) are needed to cover all the required data requirements, leading to a higher social welfare; while without data reuse, a larger set of users are needed to cover all the required data *multiple times*, leading to a lower social welfare. Intuitively, without data reuse, the number of "effective" users in the high task similarity (i.e., those can sense high popularity data only) is fewer than that in the low task similarity (i.e., those who can sense any data item), hence the social welfare becomes smaller with a high task similarity.

From Fig. 3, we can further see that with data reuse, both the social welfare achieved in the fractional VCG auction and the social welfare achieved in the randomized auction increase from 300% to 1300% when the task similarity increases from $\mu = 0$ to $\mu = 3$, comparing with those without data reuse.

V. Conclusion

In this work, we proposed a novel three-layer data-centric crowdsensing model, which captures both the task similarity and the user heterogeneity, and enables data reuse among tasks. We focused on the joint task selection and user scheduling problem, aiming at maximizing the social welfare, which is NP-hard. To address both the complexity and private information issues, we proposed a two-sided randomized auction mechanism, which is computationally efficient, individually rational, and incentive compatible (truthful) in expectation. For the future work, we are interested in achieving the budget-balanced property in the proposed randomized auction, which is essential for the MCS platform.

References

[1] R. K. Ganti, *et al.*, "Mobile crowdsensing: current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32-39, Nov. 2011.
[2] T. Das, *et al.*, "PRISM: platform for remote sensing using smartphones," in *Proc. ACM MobiSys*, June 2010.
[3] M.-R. Ra, *et al.*, "Medusa: a programming framework for crowd-sensing applications," in *Proc. ACM MobiSys*, June 2012.
[4] L. Duan, *et al.*, "Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing," *IEEE INFOCOM*, Mar. 2012.
[5] D. Yang, *et al.*, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *Proc. ACM MOBICOM*, Aug. 2012.
[6] T. Luo and C.-K. Tham, "Fairness and social welfare in incentivizing participatory sensing," in *Proc. IEEE SECON*, June 2012.
[7] Z. Feng, *et al.*, "TRAC: truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *Proc. IEEE INFOCOM*, Apr. 2014.
[8] C. Jiang, L. Gao, L. Duan, and J. Huang, "Economics of Peer-to-Peer Mobile Crowdsensing," in *Proc. IEEE GLOBECOM*, 2015.
[9] L. Gao, F. Hou, and J. Huang, "Providing long-term participation incentive in participatory sensing," in *Proc. IEEE INFOCOM*, Apr. 2015.
[10] S. Dobzinski *et al.*, "On the power of randomization in algorithmic mechanism design," *SIAM J. Comput.*, vol. 42, no. 6, Dec. 2013.
[11] R. Lavi and C. Swamy, "Truthful and near-optimal mechanism design via linear programming" in *Proc. IEEE FOCS*, Oct. 2005.
[12] L. Zhang, *et al.*, "Dynamic resource provisioning in cloud computing: a randomized auction approach," in *Proc. IEEE INFOCOM*, Apr. 2014.
[13] L. Breslau, *et al.*, "Web caching and Zipf-like distributions: evidence and implications," in *Proc. IEEE INFOCOM*, Mar. 1999.
[14] N. Nisan and A. Ronen, "Computationally feasible VCG mechanisms," *J. Artif. Intell. Res.*, vol. 29, pp.19-47, May 2007.
[15] A. Schrijver, "Theory of linear and integer programming," Wiley, 1998.