How to Crowdsource Tasks Truthfully without Sacrificing Utility: Online Incentive Mechanisms with Budget Constraint

Dong Zhao[†,‡], Xiang-Yang Li[‡], and Huadong Ma[†]

† Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia,
Beijing University of Posts and Telecommunications, China
‡ Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA
Email: dzhao@bupt.edu.cn; xli@cs.iit.edu; mhd@bupt.edu.cn

Abstract—**Mobile crowdsourced sensing (MCS) is a new paradigm which takes advantage of the pervasive smartphones to efficiently collect data, enabling numerous novel applications. To achieve good service quality for a MCS application, incentive mechanisms are necessary to attract more user participation. Most of existing mechanisms apply only for the *offline* scenario where all users' information are known a priori. On the contrary, we focus on a more realistic scenario where users arrive one by one *online* in a random order. Based on the *online auction* model, we investigate the problem that users submit their private types to the crowdsourcer when arrive, and the crowdsourcer aims at selecting a subset of users before a specified deadline for maximizing the value of the services (assumed to be a non-negative monotone submodular function) provided by selected users under a budget constraint. We design two *online mechanisms*, *OMZ* and *OMG*, satisfying the *computational efficiency*, *individual rationality*, *budget feasibility*, *truthfulness*, *consumer sovereignty* and *constant competitiveness* under the zero arrival-departure interval case and a more general case, respectively. Through extensive simulations, we evaluate the performance and validate the theoretical properties of our online mechanisms.**

I. INTRODUCTION

Crowdsourcing is a distributed problem-solving model in which a crowd of undefined size is engaged to solve a complex problem through an open call [1]. Nowadays, the proliferation of smartphones provides a new opportunity for extending existing web-based crowdsourcing applications to a larger contributing crowd, making contribution easier and omnipresent. Furthermore, today's smartphones are programmable and come with a rich set of cheap powerful embedded sensors, such as GPS, WiFi/3G/4G interfaces, accelerometer, digital compass, gyroscope, microphone, and camera. The great potential of the mobile phone sensing offers a variety of novel, efficient ways to opportunistically collect data, enabling numerous *mobile crowdsourced sensing* (MCS) applications, such as Sensorly [2] for constructing cellular/WiFi network coverage maps, SignalGuru [3], Nericell [4] and VTrack [5] for providing traffic information, Ear-Phone [6] and Noise-Tube [7] for making noise maps. For more details, we refer interested readers to several survey papers [1], [8], [9].

Adequate user participation is one of the most critical factors determining whether a MCS application can achieve good service quality. Most of the current MCS applications [2]–[7] are based on voluntary participation. While participating in a MCS campaign, smartphone users consume their own resources such as battery and computing power, and expose their locations with potential privacy threats. Thus, incentive mechanisms are necessary to provide participants with enough rewards for their participation costs. Most of existing mechanisms [10]–[14] apply only for the *offline* scenario in which all of participating users report their types, including the tasks they can complete and the bids, to the crowdsourcer (campaign organizer) in advance, and then the crowdsourcer selects a subset of users after collecting the information of all users to maximize its utility (e.g., the total value of all tasks that can be completed by selected users). In practice, however, users always arrive one by one *online* in a random order and user availability changes over time. Therefore, an *online incentive mechanism* is necessary to make irrevocable decisions on whether to accept a user's task and bid, based solely on the information of users arriving before the present moment, without knowing future information.

In this paper we consider a general problem: the crowdsourcer aims at selecting a subset of users before a specified deadline, so that the value of the services provided by selected users is maximized under the condition that the total payment to these users does not exceed a budget constraint. Specially, we investigate the case where the value function of selected users is non-negative monotone submodular. This covers many realistic scenarios. For example, many MCS applications [2]–[7] aim at selecting users to collect sensing data so that a given region can be covered before a specified deadline, where the coverage function is typically non-negative monotone submodular. We further assume that the cost and arrival/departure time of each user are private and only known to itself. Users are assumed to be game-theoretic and seek to make strategy (possibly report an untruthful cost or arrival/departure time) to maximize their individual utility in equilibrium. Thus, the problem of selecting crowdsourcing users while maximizing the value can be modeled as an *online auction*.

Our objective is to design online mechanisms satisfying six desirable properties: *computational efficiency*, *individual ra-*

tionality, *budget feasibility*, *truthfulness*, *consumer sovereignty* and *constant competitiveness*. Informally, *computational efficiency* ensures the mechanism can run in real time, *individual rationality* ensures each participating user has a non-negative utility, *budget feasibility* ensures the crowdsourcer's budget constraint is not violated, *truthfulness* ensures the participating users report their true costs (*cost-truthfulness*) and arrival/departure times (*time-truthfulness*), *consumer sovereignty* ensures each participating user has a chance to win the auction, and *constant competitiveness* guarantees that the mechanism performs close to the optimal solution in the offline scenario where all users' information are known a priori.

The main idea behind our online mechanisms is to adopt a multiple-stage sampling-accepting process. At every stage the mechanism allocates tasks to a user only if its marginal density is not less than a certain density threshold that has been computed using previous users' information, and the budget allocated for the current stage has not been exhausted. Meanwhile, the user obtains a bid-independent payment. The density threshold is computed in a manner that guarantees desirable performance properties of the mechanism. We firstly consider the *zero arrival-departure interval* case where the arrival time of each user equals to its departure time (Section III). In this case, achieving time-truthfulness is trivial. We present an online mechanism *OMZ* satisfying all desirable properties under this special case without considering the time-truthfulness. Then we revise the *OMZ* mechanism, and present another online mechanism *OMG* satisfying all desirable properties under the *general* case (Section IV).

II. System Model and Problem Formulation

We use Fig. 1 to illustrate a MCS system. The system consists of a *crowdsourcer*, which resides in the cloud and consists of multiple sensing servers, and many smartphone *users*, which are connected to the cloud by cellular networks (e.g., GSM/3G/4G) or Wi-Fi connections. The crowdsourcer first publicizes a MCS campaign in a *Region of Interest (RoI)*, aiming to finding some users to complete a set of tasks $\Gamma = \{\tau_1, \tau_2, \ldots, \tau_m\}$ in the RoI before a specified deadline T. Assume that a crowd of smartphone users $\mathcal{U} = \{1, 2, \ldots, n\}$ interested in participating in the campaign arrive online in a random order, where n is unknown. Each user i has an arrival time $a_i \in \{1, \ldots, T\}$, a departure time $d_i \in \{1, \ldots, T\}$, $d_i \geq a_i$, and a subset of tasks $\Gamma_i \subseteq \Gamma$ it can complete within this time interval. Meanwhile, user i also has an associated cost $c_i \in \mathbb{R}_+$ for performing sensing tasks. All information constitutes the *type* of user i, $\theta_i = (a_i, d_i, \Gamma_i, c_i)$. In this paper we consider an i.i.d. model which assumes that the costs and values of users are i.i.d. sampled from some *unknown* distributions. Note that the i.i.d. model is different from the *secretary* model [*] and the *oblivious adversarial model* [†].

[*]In the *secretary* model, an adversary gets to decide on the costs and values of users, but not on the *order* in which they are presented to the crowdsourcer. We further consider the *secretary* model in our online technical report [15].

[†]In the the *oblivious adversarial model*, an adversary chooses a *worst-case* input stream including the users' costs, values and their arrival orders.
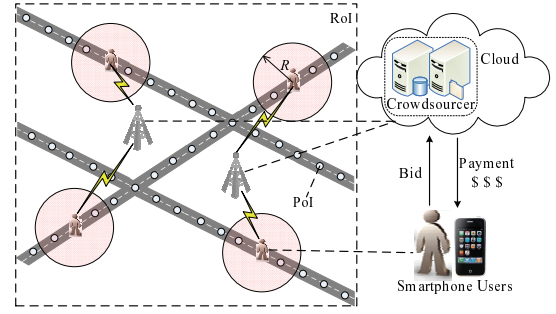

Fig. 1. Illustration of a mobile crowdsourced sensing system.

We model the interactive process between the crowdsourcer and users as an *online auction*. Each user makes a *reserve price*, called *bid*, for selling its service. When a user arrives, the crowdsourcer must decide whether to buy the service of this user, and if so, at what price, before it departs. Assume that the crowdsourcer has a budget constraint B indicating the maximum value that it is willing to pay. Therefore, the crowdsourcer always expects to obtain the maximum value from the selected users' services under the budget constraint.

Users are assumed to be game-theoretic and seek to make *strategy* to maximize their individual utility in equilibrium. Note that the arrival/departure time and cost of user i are private and only known to itself. Only the task set Γ_i must be true since the crowdsourcer can identify whether the announced tasks are performed. In other words, user i may misreport all information about its type except for Γ_i. The budget and value function of the crowdsourcer are common knowledge. Although we do not require a user to declare its departure time until the moment of its departure, we find it convenient to analyze our auctions as direct-revelation mechanisms (DRMs). The strategy space in an online DRM allows a user to declare some possibly untruthful type $\hat{\theta}_i = (\hat{a}_i, \hat{d}_i, \Gamma_i, b_i)$, subject to $a_i \leq \hat{a}_i \leq \hat{d}_i \leq d_i$. Note that we assume that a user cannot announce an earlier arrival time or a later departure time than its true arrival/departure time. In order to obtain the required service, the crowdsourcer needs to design an *online mechanism* $\mathcal{M} = (f, p)$ consisting of an *allocation* function f and a *payment* function p. For any *strategy sequence* $\hat{\theta} = (\hat{\theta}_1, \ldots, \hat{\theta}_n)$, the allocation function $f(\hat{\theta})$ computes an allocation of tasks for a selected subset of users $\mathcal{S} \subseteq \mathcal{U}$, and the payment function $p(\hat{\theta})$ returns a vector $(p_1(\hat{\theta}), \ldots, p_n(\hat{\theta}))$ of payments to the users. Note that, the crowdsourcer, when presented with the strategy $\hat{\theta}_i$ of user i, must decide whether to accept user i at what price (p_i) before the time step $\hat{d}_i$.

The *utility* of user i is

$$u_i = \begin{cases} p_i - c_i, & \text{if } i \in \mathcal{S}; \\ 0, & \text{otherwise.} \end{cases}$$

Let $V(\mathcal{S})$ denote the *value* function of the crowdsourcer over the selected subset of users $\mathcal{S}$. The crowdsourcer expects to obtain the maximum value from the selected users' services under the budget constraint, i.e.,

$$\textbf{Maximize } V(\mathcal{S}) \textbf{ subject to } \sum_{i \in \mathcal{S}} p_i \leq B.$$

In this paper, we focus on the case where $V(\mathcal{S})$ is non-negative monotone submodular. This covers many realistic scenarios.

Definition 1 (Monotone Submodular Function)**.** *Let Ω be a finite set. For any $X \subseteq Y \subseteq \Omega$ and $x \in \Omega \backslash Y$, a function $f : 2^\Omega \mapsto \mathbb{R}$ is called submodular if and only if*

$$f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y),$$

and it is monotone (increasing) if and only if $f(X) \leq f(Y)$.

An Application Example: As illustrated in Fig. 1, we consider the scenario where the crowdsourcer expects to obtain the sensing data covering all roads in a RoI. For convenience of calculations, we divide each road in the RoI into multiple discrete *Points of Interest (PoIs)*, and the objective of the crowdsourcer is equivalent to obtaining the sensing data covering all PoIs before T. The set of PoIs is denoted by $\Gamma = \{\tau_1, \tau_2, \ldots, \tau_m\}$. Assume that each sensor follows a geometric disk sensing model with sensing range R, which means if user i senses at a location L_i and obtain a reading, then any PoI within the disk with the origin at L_i and a radius of R has been covered once. The set of PoIs covered by user i is denoted by $\Gamma_i \subseteq \Gamma$, which means the sensing tasks that user i can complete. Without loss of generality, assume that each PoI τ_j has a coverage requirement $r_j \in \mathbb{Z}_+$ indicating how many times it requires to be sensed at most. The *value* of the selected users to the crowdsourcer is:

$$V(\mathcal{S}) = \sum_{j=1}^{m} \min\{r_j, \sum_{i \in \mathcal{S}} v_{i,j}\},$$

where $v_{i,j}$ equals to 1 if $\tau_j \in \Gamma_i$, and 0 otherwise. $V(\mathcal{S})$ is non-negative monotone submodular, the proof of which is given in our online technical report [15] due to page limit.

Our objective is to design online mechanisms satisfying the following six desirable properties:

- **Computational Efficiency:** A mechanism is *computationally efficient* if both the allocation and payment can be computed in polynomial time as each user arrives.
- **Individual Rationality:** Each participating user will have a non-negative utility: $u_i \geq 0$.
- **Budget Feasibility:** We require the mechanism to be *budget feasible*: $\sum_{i \in \mathcal{S}} p_i \leq B$.
- **Truthfulness:** A mechanism is *cost-* and *time-truthful* (or simply called *truthful*, or *incentive compatible* or *strategyproof*) if reporting the true cost and arrival/departure time is a *dominant strategy* for all users. In other words, no user can improve its utility by submitting a false cost or arrival/departure time, no matter what others submit.
- **Consumer Sovereignty:** The mechanism cannot arbitrarily exclude a user; each user will have a chance to win the auction and obtain a payment if only its bid is sufficiently low while others are fixed.
- **Competitiveness:** The goal of the mechanism is to maximize the value of the crowdsourcer. To quantify the performance of the mechanism we compare its solution with the *optimal solution* that can be obtained

in the offline scenario where the crowdsourcer has full knowledge about users' types. A mechanism is $O(g(n))$-*competitive* if the ratio between the online solution and the optimal solution is $O(g(n))$. Ideally, we would like our mechanism to be $O(1)$-*competitive*.

The importance of the first three properties is obvious, because they together guarantee that the mechanisms can be implemented in real time and satisfy the basic requirements. In addition, the last three properties are indispensable for guaranteeing the high performance and robustness. The *truthfulness* aims to eliminate the fear of market manipulation and the overhead of strategizing over others for the participating users. The *consumer sovereignty* aims to guarantee that each participating user has a chance to win the auction, otherwise it will hinder the users' competition or even result in task starvation. Besides, if some users are guaranteed not to win the auction, then being truthful or not will have the same outcome. For this reason, the property satisfying both the *consumer sovereignty* and *truthfulness* is also called *strong truthfulness* by Hajiaghayi et al. [16]. Later we will show that satisfying the consumer sovereignty is not trivial in the online scenario, which is in contrast to the offline scenario.

III. ONLINE MECHANISM UNDER ZERO ARRIVAL-DEPARTURE INTERVAL CASE

In this section, we consider a special case where the arrival time of each user equals to its departure time. In this case, each user is impatient since the decision must be made immediately once it arrives. Note that achieving time-truthfulness is trivial in this case. It is because that any user has no incentive to report a later arrival time or an earlier departure time than its true arrival/departure time, since the user cannot perform any sensing task or obtain a payment after it departs. We present an online mechanism *OMZ* satisfying all desirable properties under this *zero arrival-departure interval* case, without considering the time-truthfulness. To facilitate understanding, it is also assumed that no two users have the same arrival time. Note that this assumption can also be easily removed according to the revised mechanism in Section IV.

A. Mechanism Design

An online mechanism needs to overcome several nontrivial challenges: firstly, the users' costs are unknown and need to be reported in a truthful manner; secondly, the total payment cannot exceed the crowdsourcer's budget; finally, the mechanism needs to cope with the online arrival of users. Previous solutions of online auctions and related problems [17], [18] always achieve desirable outcomes in online settings via a two-stage sampling-accepting process: the first batch of users is rejected and used as the sample which enables making an informed decision on whether accepting the rest of users. However, these solutions cannot guarantee the *consumer sovereignty*, since the first batch of users has no chance to win the auction no matter how low its cost is. It can lead to undesirable effects in our problem: automatically rejecting the first batch of users encourages users to arrive late; in other
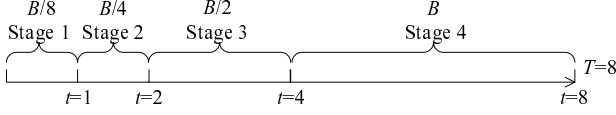
Fig. 2. Illustration of a multiple-stage sampling-accepting process when $T = 8$.

words, those users arriving early have no incentive to report their bids, which may hinder the users' competition or even result in task starvation.

To address the above challenges, we design our online mechanism, *OMZ*, based on a *multiple-stage sampling-accepting* process. The mechanism dynamically increases the sample size and learns a *density threshold* used for future decision, while increasing the *stage-budget* it uses for allocation at various stages. The whole process is illustrated in Algorithm 1. Firstly, we divide all of T time steps into $(\lfloor \log_2 T \rfloor + 1)$ stages: $\{1, 2, \ldots, \lfloor \log_2 T \rfloor, \lfloor \log_2 T \rfloor + 1\}$. The stage i ends at time step $T' = \lfloor 2^{i-1}T/2^{\lfloor \log_2 T \rfloor} \rfloor$. Correspondingly, the stage-budget for the i-th stage is allocated as $B' = 2^{i-1}B/2^{\lfloor \log_2 T \rfloor}$. Fig. 2 is an illustration when $T = 8$. When a stage is over, we add all users who have arrived into the sample set $\mathcal{S}'$, and compute a density threshold ρ^* according to the information of samples and the allocated stage-budget B'. This density threshold is computed by calling the **GetDensityThreshold** algorithm (to be elaborated later), and used for making decision at the next stage. Specially, when the last stage $i = \lfloor \log_2 T \rfloor + 1$ comes, the density threshold has been computed according to the information of all users arriving before time step $\lfloor T/2 \rfloor$, and the allocated stage-budget $B/2$.

Algorithm 1: Online Mechanism under Zero Arrival-departure Interval Case (*OMZ*)

Input: Budget constraint B, deadline T
1   $(t, T', B', \mathcal{S}', \rho^*, \mathcal{S}) \leftarrow (1, \frac{T}{2^{\lfloor \log_2 T \rfloor}}, \frac{B}{2^{\lfloor \log_2 T \rfloor}}, \emptyset, \epsilon, \emptyset)$;
2   **while** $t \leq T$ **do**
3   **if** *there is a user i arriving at time step t* **then**
4   **if** $b_i \leq V_i(\mathcal{S})/\rho^* \leq B' - \sum_{j \in \mathcal{S}} p_j$ **then**
5         $p_i \leftarrow V_i(\mathcal{S})/\rho^*$; $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$;
6   **else** $p_i \leftarrow 0$;
7       $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{i\}$;
8   **end**
9   **if** $t = \lfloor T' \rfloor$ **then**
10       $\rho^* \leftarrow$ **GetDensityThreshold**$(B', \mathcal{S}')$;
11       $T' \leftarrow 2T'$; $B' \leftarrow 2B'$;
12   **end**
13     $t \leftarrow t + 1$;
14 **end**

Given a set of selected users $\mathcal{S}$, the *marginal value* of user $i \notin \mathcal{S}$ is $V_i(\mathcal{S}) = V(\mathcal{S} \cup \{i\}) - V(\mathcal{S})$, and its *marginal density* is $V_i(\mathcal{S})/b_i$. When a new user i arrives, the mechanism allocates tasks to it as long as its marginal density is not less than the current threshold density ρ^*, and the allocated stage-budget B' has not been exhausted. Meanwhile, we give user i a payment

$$p_i = V_i(\mathcal{S})/\rho^*,$$

and add this user to the set of selected users $\mathcal{S}$. To start the

mechanism, we initially set a small density threshold ϵ, which is used for making decision at the first stage.

Since each stage maintains a common density threshold, it is natural to adopt a *proportional share* allocation rule to compute the density threshold from the sample set $\mathcal{S}'$ and the allocated stage-budget B'. As illustrated in Algorithm 2, the computation process adopts a greedy strategy. Users are sorted according to their increasing marginal densities. In this sorting the $(i+1)$-th user is the user j such that $V_j(\mathcal{J}_i)/b_j$ is maximized over $\mathcal{S}' \backslash \mathcal{J}_i$, where $\mathcal{J}_i = \{1, 2, \ldots, i\}$ and $\mathcal{J}_0 = \emptyset$. Considering the submodularity of V, this sorting implies that:

$$\frac{V_1(\mathcal{J}_0)}{b_1} \geq \frac{V_2(\mathcal{J}_1)}{b_2} \geq \cdots \geq \frac{V_{|\mathcal{S}'|}(\mathcal{J}_{|\mathcal{S}'|-1})}{b_{|\mathcal{S}'|}}.$$

Find the largest k such that $b_k \leq \frac{V_k(\mathcal{J}_{k-1})B}{V(\mathcal{J}_k)}$. The set of selected users is $\mathcal{J}_k = \{1, 2, \ldots, k\}$. Finally, we set the density threshold to be $\frac{V(\mathcal{J}_k)}{\delta B'}$. Here we set $\delta > 1$ to obtain a slight underestimate of the density threshold for guaranteeing enough users selected and avoiding the waste of budget. Later we will fix the value of δ elaborately to enable the mechanism achieving a constant competitive ratio.

Algorithm 2: GetDensityThreshold

Input: Stage-budget B', sample set $\mathcal{S}'$
1   $\mathcal{J} \leftarrow \emptyset$; $i \leftarrow \arg\max_{j \in \mathcal{S}'}(V_j(\mathcal{J})/b_j)$;
2   **while** $b_i \leq \frac{V_i(\mathcal{J})B'}{V(\mathcal{J} \cup \{i\})}$ **do**
3     $\mathcal{J} \leftarrow \mathcal{J} \cup \{i\}$;
4     $i \leftarrow \arg\max_{j \in \mathcal{S}' \backslash \mathcal{J}}(V_j(\mathcal{J})/b_j)$;
5   **end**
6   $\rho \leftarrow V(\mathcal{J})/B'$;
7   **return** ρ/δ;

B. Mechanism Analysis

In the following, we will firstly prove that *OMZ* satisfies the computational efficiency (Lemma 1), individual rationality (Lemma 2), budget feasibility (Lemma 3), cost-truthfulness (Lemma 4), and consumer sovereignty (Lemma 5). Then, we will prove that *OMZ* can achieve a constant competitive ratio by elaborately fixing the value of δ (Lemma 6).

Lemma 1. *The OMZ mechanism is computationally efficient.*

Proof: Since the mechanism runs online, we only need to focus on the computation complexity at each time step $t \in \{1, \ldots, T\}$. Computing the marginal value of user i takes $O(|\Gamma_i|)$ time, which is at most $O(m)$. Thus, the running time of computing the allocation and payment of user i (lines 3-8) is bounded by $O(m)$. Next, we analyze the complexity of computing the density threshold (Algorithm 2). Finding the user with maximum marginal density takes $O(m|\mathcal{S}'|)$ time. Since there are m tasks and each selected user should contribute at least one new task, the number of winners is at most $\min\{m, |\mathcal{S}'|\}$. Thus, the running time of Algorithm 2 is bounded by $O(m|\mathcal{S}'|\min\{m, |\mathcal{S}'|\})$. Therefore, the computation complexity at each time step (lines 3-13) is bounded by $O(m|\mathcal{S}'|\min\{m, |\mathcal{S}'|\})$. At the last stage, the sample set $\mathcal{S}'$

has the maximum number of samples, being $n/2$ with high probability. Thus, the computation complexity at each time step is bounded by $O(mn\min\{m,n\})$. ∎

Note that the above analysis is very conservative. In practice, the running time $O(|\Gamma_i|)$ of computing the marginal value is much less than $O(m)$. Moreover, the running time of *OMZ* will increase linearly with n especially when n is large.

Lemma 2. *The OMZ mechanism is individually rational.*

Proof: From the lines 4-6 of Algorithm 1, we know that $p_i \geq b_i$ if $i \in \mathcal{S}$, otherwise $p_i = 0$. Thus, we have $u_i \geq 0$. ∎

Lemma 3. *The OMZ mechanism is budget feasible.*

Proof: At each stage $i \in \{1, 2, \ldots, \lfloor \log_2 T \rfloor, \lfloor \log_2 T \rfloor + 1\}$, the mechanism uses a stage-budget $B' = \frac{2^{i-1}B}{2^{\lfloor \log_2 T \rfloor}}$. From the lines 4-5 of Algorithm 1, it is guaranteed that the current total payment does not exceed the stage-budget B'. Specially, the budget constraint of the last stage is B. Therefore, every stage is budget feasible, and when the deadline T arrives, the total payment does not exceed B. ∎

Designing a cost-truthful mechanism relies on the rationale of *bid-independence*. Let b_{-i} denote the sequence of bids arriving before the i-th bid b_i, i.e., $b_{-i} = (b_1, \ldots, b_{i-1})$. We call such a sequence *prefixal*. Let p' be a function from prefixal sequences to prices (non-negative real numbers). We extend the definition of bid-independence [19] to the online scenario:

Definition 2 (Bid-independent Online Auction). *An online auction is called bid-independent if the allocation and payment rules for each player i satisfy:*

a) *The auction constructs a price schedule $p'(b_{-i})$;*
b) *If $p'(b_{-i}) \geq b_i$, player i wins at price $p_i = p'(b_{-i})$;*
c) *Otherwise, player i is rejected, and $p_i = 0$.*

Proposition 1. *([20], Proposition 2.1) An online auction is cost-truthful if and only if it is bid-independent.*

Lemma 4. *The OMZ mechanism is cost-truthful.*

Proof: Consider a user i that arrives at some stage for which the density threshold is ρ^*. If by the time the user arrives there are no remaining budget, then the user's cost declaration will not affect the allocation of the mechanism and thus cannot improve its utility by submitting a false cost. Otherwise, assume there are remaining budget by the time the user arrives. In case $c_i \leq V_i(\mathcal{S})/\rho^*$, reporting any cost below $V_i(\mathcal{S})/\rho^*$ would not make a difference in the user's allocation and payment and its utility would be $V_i(\mathcal{S})/\rho^* - c_i \geq 0$. Declaring a cost above $V_i(\mathcal{S})/\rho^*$ would make the worker lose the auction, and its utility would be 0. In case $c_i > V_i(\mathcal{S})/\rho^*$, declaring any cost above $V_i(\mathcal{S})/\rho^*$ would leave the user unallocated with utility 0. If the user declares a cost lower than $V_i(\mathcal{S})/\rho^*$ it will be allocated. In such a case, however, its utility will be negative. Hence the user's utility is always maximized by reporting its true cost: $b_i = c_i$. ∎

Lemma 5. *The OMZ mechanism satisfies the consumer sovereignty.*

Proof: Each stage is an accepting process as well as a sampling process ready for the next stage. As a result, users are not automatically rejected during the sampling process, and are allocated as long as their marginal densities are not less than the current density threshold, and the allocated stage-budget has not been exhausted. ∎

Before analyzing the competitiveness of the *OMZ* mechanism, we firstly introduce an offline mechanism proposed by Singer [21], which has been proved to satisfy the *computational efficiency*, *individual rationality*, *budget feasibility*, and *truthfulness*. This mechanism does not have knowledge about users' costs, but it is an offline mechanism, i.e., all users submit their bids to the mechanism and wait for the mechanism to collect all the bids and decide on an allocation. This mechanism has been proved to be $O(1)$-*competitive* in maximizing the value of services received under budget constraint compared with the optimal solution. Therefore, we only need to prove that *OMZ* has a constant competitive ratio compared with this offline mechanism, then *OMZ* will also be $O(1)$-*competitive* compared with the optimal solution. Note that in the offline scenario satisfying the *time-truthfulness* and *consumer sovereignty* is trivial, since all decisions are made after all users' information is submitted to the crowdsourcer.

Algorithm 3: Proportional Share Mechanism (Offline) [21]

Input: Budget constraint B, User set $\mathcal{U}$
  /* Winner selection phase   */
1 $\mathcal{S} \leftarrow \emptyset$; $i \leftarrow \arg\max_{j \in \mathcal{U}}(V_j(\mathcal{S})/b_j)$;
2 **while** $b_i \leq \frac{V_i(\mathcal{S})B}{V(\mathcal{S} \cup \{i\})}$ **do**
3     $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$;
4     $i \leftarrow \arg\max_{j \in \mathcal{U} \setminus \mathcal{S}}(V_j(\mathcal{S})/b_j)$;
5 **end**
  /* Payment determination phase   */
6 **foreach** $i \in \mathcal{U}$ **do** $p_i \leftarrow 0$;
7 **foreach** $i \in \mathcal{S}$ **do**
8     $\mathcal{U}' \leftarrow \mathcal{U} \setminus \{i\}$; $\mathcal{Q} \leftarrow \emptyset$;
9   **repeat**
10       $i_j \leftarrow \arg\max_{j \in \mathcal{U}' \setminus \mathcal{Q}}(V_j(\mathcal{Q})/b_j)$;
11       $p_i \leftarrow \max\{p_i, \min\{b_{i(j)}, \eta_{i(j)}\}\}$;
12   **until** $b_{i_j} \leq \frac{V_{i_j}(\mathcal{Q}_{j-1})B}{V(\mathcal{Q})}$;
13 **end**
14 **return** $(\mathcal{S}, p)$;

The offline mechanism adopts a *proportional share allocation rule*. As described in Algorithm 3, it consists of two phases: the *winner selection* phase and the *payment determination* phase. The *winner selection* phase has the same working process as Algorithm 2. To compute the payment for each winner $i \in \mathcal{S}$, we sort the users in $\mathcal{U} \setminus \{i\}$:

$$\frac{V_{i_1}(\mathcal{Q}_0)}{b_{i_1}} \geq \frac{V_{i_2}(\mathcal{Q}_1)}{b_{i_2}} \geq \cdots \geq \frac{V_{i_{n-1}}(\mathcal{Q}_{n-2})}{b_{i_{n-1}}},$$

where $V_{i_j}(\mathcal{Q}_{j-1}) = V(\mathcal{Q}_{j-1} \cup \{i_j\}) - V(\mathcal{Q}_{j-1})$ denotes the marginal value of the j-th user and $\mathcal{Q}_j$ denotes the first j users according to this sorting over $\mathcal{U} \setminus \{i\}$ and $\mathcal{Q}_0 = \emptyset$. The marginal value of user i at position j is $V_{i(j)}(\mathcal{Q}_{j-1}) = V(\mathcal{Q}_{j-1} \cup \{i\}) - V(\mathcal{Q}_{j-1})$. Let k' denote the position of the last user $i_j \in \mathcal{U} \setminus \{i\}$, such that $b_{i_j} \leq V_{i_j}(\mathcal{Q}_{j-1})B/V(\mathcal{Q}_j)$. For

brevity we will write $b_{i(j)} = V_{i(j)}(\mathcal{Q}_{j-1})b_{i_j}/V_{i_j}(\mathcal{Q}_{j-1})$, and $\eta_{i(j)} = V_{i(j)}(\mathcal{Q}_{j-1})B/V(\mathcal{Q}_{j-1} \cup \{i\})$. In order to guarantee the truthfulness, each winner should be paid the critical value, which means that user i would not win the auction if it bids higher than this value. Thus, the payment for user i should be the maximum of these $k'+1$ prices:

$$p_i = \max_{j \in [k'+1]} \{\min\{b_{i(j)}, \eta_{i(j)}\}\}.$$

Let Z be the set of selected users $\mathcal{S}$ computed by Algorithm 3, and the value of Z is $V(Z)$. The density of Z is $\rho = V(Z)/B$. Define Z_1 and Z_2 as the subsets of Z that appears in the first and second half of the input stream, respectively. When the stage $\lfloor \log_2 T \rfloor$ is over, we obtain the sample set $\mathcal{S}'$ consisting of all users arriving before the time $\lfloor T/2 \rfloor$. Thus, we have $Z_1 = Z \cap \mathcal{S}'$, and $Z_2 = Z \cap \{\mathcal{U} \backslash \mathcal{S}'\}$. Let Z_1' denote the set of selected users computed by Algorithm 2 based on the sample set $\mathcal{S}'$ and the allocated stage-budget $B/2$, and the value of Z_1' is $V(Z_1')$. The density of Z_1' is $\rho_1' = 2V(Z_1')/B$. The density threshold of the last stage is $\rho^* = \rho_1'/\delta$. Let Z_2' denote the set of selected users computed by Algorithm 1 at the last stage. Assume that the value of each user is at most $V(Z)/\omega$, where the parameter ω will be fixed later.

Since the costs and values of all users are i.i.d., they can be selected in the set Z with the same probability. The sample set $\mathcal{S}'$ is a random subset of $\mathcal{U}$ since all users arrive in a random order. Therefore the number of users from Z in the sample set $\mathcal{S}'$ follows a hypergeometric distribution $H(n/2, |Z|, n)$. Thus, we have $\mathbb{E}[|Z_1|] = \mathbb{E}[|Z_2|] = |Z|/2$. The value of each user can be seen as an i.i.d. random variable, and because of the submodularity of $V(\mathcal{S})$, it can be derived that: $\mathbb{E}[V(Z_1)] = \mathbb{E}[V(Z_2)] \geq V(Z)/2$. The expected total payments to the users from both Z_1 and Z_2 are $B/2$. Since $V(Z_1')$ is computed with the stage-budget $B/2$, it can be derived that: $\mathbb{E}[V(Z_1')] \geq \mathbb{E}[V(Z_1)] \geq V(Z)/2$, and $\mathbb{E}[\rho_1'] \geq \rho$, where the first inequality follows from the fact that $V(Z_1')$ is the optimal solution computed by Algorithm 2 with stage-budget $B/2$ according to the *proportional share allocation* rule. Therefore, we only need to prove that the ratio of $\mathbb{E}[V(Z_2')]$ to $\mathbb{E}[V(Z_1')]$ is at least a constant, then the *OMZ* mechanism will also have a constant expected competitive ratio compared with the offline mechanism.

Lemma 6. *For sufficiently large ω, the ratio of $\mathbb{E}[V(Z_2')]$ to $\mathbb{E}[V(Z_1')]$ is at least a constant. Specially, this ratio approaches $1/4$ as $\omega \to \infty$ and $\delta \to 4$.*

The proof of Lemma 6 is given in Appendix A.

Theorem 1. *The OMZ mechanism satisfies the computational efficiency, individual rationality, budget feasibility, truthfulness, consumer sovereignty, and constant competitiveness under the zero arrival-departure interval case.*

IV. ONLINE MECHANISM UNDER GENERAL CASE

In this section, we consider the general case where each user may have a *non-zero arrival-departure interval*, and there may be multiple online users in the auction simultaneously.

A. Mechanism Design

In order to hold desirable properties of *OMZ*, we adopt a similar algorithm framework under the general case. Meanwhile, in order to guarantee the *cost-* and *time-truthfulness*, it is necessary to modify *OMZ* based on three principles. Firstly, any user is added to the sample set only when it departs; otherwise, the bid-independence will be destroyed if its arrival-departure time spans multiple stages, because a user can indirectly affect its payment now. Secondly, if there are multiple users who have not yet departed at some time, we sort these online users according to their marginal values, instead of marginal densities, and preferentially select those users with higher marginal value. In this way, the bid-independence can be held. Thirdly, whenever a new time step arrives, it scans through the list of users who have not yet departed and selects those whose marginal densities are not less than the current density threshold under the stage-budget constraint, even if some arrived much earlier. At the departure time of any user who was selected as a winner, the user is paid for a price equal to the maximum price attained during the user's reported arrival-departure interval, even if this price is larger than the price at the time step when the user was selected as a winner.

Algorithm 4: Online Mechanism under General Case (*OMG*)

Input: Budget constraint B, deadline T
1   $(t, T', B', \mathcal{S}', \rho^*, \mathcal{S}) \leftarrow (1, \frac{T}{2^{\lfloor \log_2 T \rfloor}}, \frac{B}{2^{\lfloor \log_2 T \rfloor}}, \emptyset, \epsilon, \emptyset)$;
2 **while** $t \leq T$ **do**
3   Add all new users arriving at time step t to a set of online users $\mathcal{O}$; $\mathcal{O}' \leftarrow \mathcal{O} \setminus S$;
4   **repeat**
5      $i \leftarrow \arg\max_{j \in \mathcal{O}'}(V_j(\mathcal{S}))$;
6   **if** $b_i \leq V_i(\mathcal{S})/\rho^* \leq B' - \sum_{j \in \mathcal{S}} p_j$ **then**
7       $p_i \leftarrow V_i(\mathcal{S})/\rho^*$; $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$;
8   **else** $p_i \leftarrow 0$;
9      $\mathcal{O}' \leftarrow \mathcal{O}' \setminus \{i\}$;
10   **until** $\mathcal{O}' = \emptyset$;
11   Remove all users departing at time step t from $\mathcal{O}$, and add them to $\mathcal{S}'$;
12   **if** $t = \lfloor T' \rfloor$ **then**
13      $\rho^* \leftarrow$ **GetDensityThreshold**$(B', \mathcal{S}')$;
14      $T' \leftarrow 2T'$; $B' \leftarrow 2B'$; $\mathcal{O}' \leftarrow \mathcal{O}$;
15   **repeat**
16       $i \leftarrow \arg\max_{j \in \mathcal{O}'}(V_j(\mathcal{S} \setminus \{j\}))$;
17   **if** $b_i \leq V_i(\mathcal{S} \setminus \{i\})/\rho^* \leq B' - \sum_{j \in \mathcal{S}} p_j + p_i$ **and** $V_i(\mathcal{S} \setminus \{i\})/\rho^* > p_i$ **then**
18        $p_i \leftarrow V_i(\mathcal{S} \setminus \{i\})/\rho^*$;
19   **if** $i \notin \mathcal{S}$ **then** $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$;
20   **end**
21       $\mathcal{O}' \leftarrow \mathcal{O}' \setminus \{i\}$;
22   **until** $\mathcal{O}' = \emptyset$;
23   **end**
24    $t \leftarrow t + 1$;
25 **end**

According to the above principles, we design *OMG* satisfying all desirable properties under the general case, as described in Algorithm 4. Specially, we consider two cases. The first case is when the current time step t is not at the end of any stage. In this case, the density threshold remains unchanged. The following operations (the lines 3-11 in Algorithm 4) are

performed. Firstly, all new users arriving at time step t are added to a set of online users $\mathcal{O}$. Then we make decision on whether to select these online users one by one in the order of their marginal values; the users with higher marginal values will be selected first. If an online user i has been selected as a winner before time step t, we need not to make decision on it again because it is impossible to obtain a higher payment than before (refer to our proof in [15]). Otherwise, we need to make decision on it again: if its marginal density is not less than the current density threshold, and the allocated stage-budget has not been exhausted, it will be selected as a winner. Meanwhile, we give user i a payment $p_i = V_i(\mathcal{S})/\rho^*$, and add it to the set of selected users $\mathcal{S}$. Finally, we remove all users departing at time step t from $\mathcal{O}$, and add them to the sample set $\mathcal{S}'$.

The second case is when the current time step is just at the end of some stage. In this case, the density threshold will be updated. The mechanism works as the lines 13-22. We need to make decision on whether to select these online users, and at what prices, one by one in the order of their marginal values, no matter whether they have ever been selected as the winners before time step t. As shown in the lines 17-20, if user i can obtain a higher payment than before, its payment will be updated. Meanwhile, if user i has never been selected as a winner before time step t, it will be added to the set $\mathcal{S}$.

B. Mechanism Analysis

It is convenient to prove that the *OMG* mechanism also holds the *individual rationality*, *consumer sovereignty*, and *constant competitiveness* as *OMZ* (with almost the same proof), although *OMG* may have slightly lower competitive ratio than *OMZ*. In the following, we prove that *OMG* also satisfies the *computational efficiency*, the *budget feasibility*, and most importantly, the *cost-* and *time-truthfulness*.

Lemma 7. *The OMG mechanism is computationally efficient.*

Proof: The *OMG* mechanism needs to compute the allocations and payments of multiple online users at each time step. Thus, the running time of computing the allocations and payments at each time step is bounded by $O(m|\mathcal{O}|) < O(mn)$, where $|\mathcal{O}|$ is the number of online users. The complexity of computing the density threshold is the same as that of *OMZ*. Thus, the computation complexity at each time step is the same as that of *OMZ*, i.e., bounded by $O(mn\min\{m,n\})$. ∎

Lemma 8. *The OMG mechanism is budget feasible.*

Proof: From the lines 6-7 and 17-18 of Algorithm 4, it is guaranteed that the current total payment does not exceed the stage-budget B'. Note that in the line 17, p_i is the price paid for user i in the previous stage instead of the current stage, so it cannot lead to the overrun of the current stage-budget. Thus every stage is budget feasible, and when the deadline T arrives, the total payment does not exceed B. ∎

Lemma 9. *The OMG mechanism is cost- and time-truthful.*

The proof of Lemma 9 is quite lengthy and is given in our online technical report [15].

Theorem 2. *The OMG mechanism satisfies the computational efficiency, individual rationality, budget feasibility, truthfulness, consumer sovereignty, and constant competitiveness under the general case.*

V. PERFORMANCE EVALUATION

To evaluate the performance of our online mechanisms, we implemented *OMZ* and *OMG*, and compared them against the following three benchmarks. The first benchmark is the (approximate) *optimal* offline solution which has full knowledge about all users' types. The problem in this scenario is essentially a *budgeted maximum coverage problem*, which is a well-known NP-hard problem. It is known that a greedy algorithm provides a $(1 - 1/e)$-approximation solution [22]. The second benchmark is the *proportional share* mechanism in the offline scenario (Algorithm 3). The third benchmark is the *random* mechanism, which adopts a naive strategy, i.e., rewards users based on an uninformed fixed density threshold. The performance metrics include the *running time* and the *crowdsourcer's value*.

A. Simulation Setup

We consider a Wi-Fi signal sensing application with the same scenario as [23]. The RoI is located at Manhattan, NY, including three avenues of 0.319km length and three streets of 1.135km length. We divide each road in the RoI into discrete PoIs with a uniform spacing of 1m, so the RoI consists of 4353 PoIs ($m = 4352$) in total. Without loss of generality, let the coverage requirement of each PoI be 1. We set the deadline (T) to 1800s, and vary the budget (B) from 100 to 10000 with the increment of 100. Users arrive according to a Poisson process in time with arrival rate λ. We vary λ from 0.2 to 1 with the increment of 0.2. Whenever a user arrives, it is placed at a random location on the roads. In *OMZ* each user has zero arrival-departure interval, and in *OMG* the arrival-departure interval of each user is uniformly distributed over $[0, 300]$ seconds. The sensing range (R) of each sensor is set to 7 meters. Each user's cost is uniformly distributed over $[1, 10]$. The initial density threshold (ϵ) of Algorithm 1 and 4 is set to 1. As we proved in Lemma 6, when $\delta = 4$ *OMZ* is $O(1)$-competitive for sufficiently large ω. Meanwhile we note that ω increases with the number of users who have arrived. Thus, we set $\delta = 1$ initially, and change it to $\delta = 4$ once the size of the sample set exceeds a specified threshold. Note that this threshold could be an empirical value. In our simulation, we set this threshold to 240, because we observe that each user's value is at most $1/100$ of the total value when the number of users is larger than 240. For the *random* mechanism, we obtain the average performance of 50 such solutions for evaluations, where in each solution the density threshold was chosen at random from the range of 1 to 29 [‡]. All the simulations were run on a PC with 1.7 GHz CPU and 8 GB memory. Each measurement is averaged over 100 instances.

[‡]Each user can cover at most 29 PoIs, and its bid is at least 1, so its marginal density is at most 29.
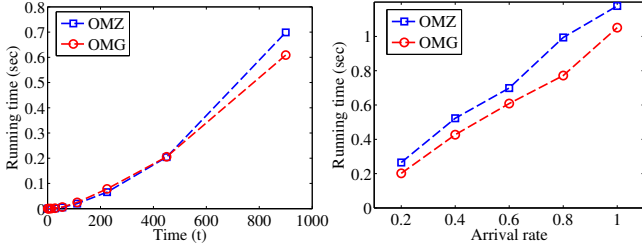
(a) At different stages ($\lambda = 0.6$)   (b) Impact of λ (at the last stage)

Fig. 3. Running Time.
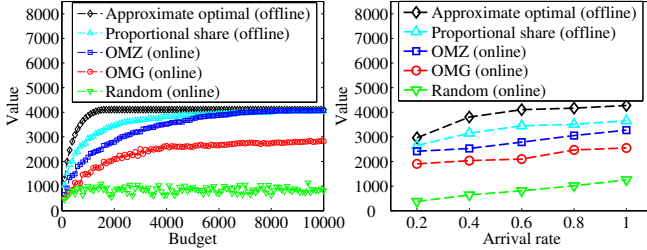
(a) Impact of B ($\lambda = 0.6$)   (b) Impact of λ ($B = 2000$)

Fig. 4. Crowdsourcer's value.

B. Evaluation Results

Running Time: Fig. 3 shows the running time of *OMZ* and *OMG*. Specially, Fig. 3(a) plots the running time at different stages while $\lambda = 0.6$[§]. Fig. 3(b) plots the running time at the last stage with different arrival rates (λ). Both the *OMZ* and *OMG* mechanisms have similar performance while *OMG* outperforms *OMZ* slightly. Note that the size of the sample set increases linearly with the time t and the arrival rate λ, so Fig. 3 implies the relationship between the running time and the number of users. Thus, from Fig. 3 we can infer that the running time increases linearly with the number of users (n), which is consistent with our analysis in Section III-B.

Crowdsourcer's Value: Fig. 4 compares the crowdsourcer's value achieved by *OMZ* and *OMG* against the three benchmarks. From Fig. 4(a) we can observe that the crowdsourcer obtains higher value when the budget constraint increases. From Fig. 4(b) we can observe that the crowdsourcer obtains higher value when more users participate. The *approximate optimal* mechanism and the *proportional share* mechanism operate in the offline scenario, where the true types or strategies of all users are known a priori, and will therefore always outperform *OMZ* and *OMG*. It is shown that the *proportional share* mechanism sacrifices some value of the crowdsourcer to achieve the cost-truthfulness compared with the *approximate optimal* mechanism, and *OMG* also sacrifices some value to achieve the time-truthfulness compared with *OMZ*. We can also observe that both *OMZ* and *OMG* are guaranteed to be within a constant factor of the offline solutions. Specially, although both *OMZ* and *OMG* are only guaranteed to be within a competitive factor of at least 8 of the *proportional share* solution in expectation as we proved in Lemma 6, the

---

[§]As we proved in Lemma 1, the computation complexity is dominated by computing the density threshold, so only the running time at the end time of each stage is plotted.

simulation results show that this ratio is almost as small as 1.6 for *OMZ* or 2.4 for *OMG*. As compared to the *approximate optimal* solution, this ratio is still below 2.2 for *OMZ* or below 3.4 for *OMG*. In addition, we can see that both *OMZ* and *OMG* largely outweigh the *random* mechanism.

VI. RELATED WORK

At present, there are some studies [10]–[14] on incentive mechanism design for MCS applications in the offline scenario. Generally, two system models are considered: the platform/crowdsourcer-centric model where the crowdsourcer provides a fixed reward to participating users, and the user-centric model where users can have their reserve prices for the sensing service. For the crowdsourcer-centric model, incentive mechanisms were designed by using a Stackelberg game [12], [13], where the costs of all users or their probability distribution was assumed to be known. In contrast, the user-centric model allows that each user has a private cost only known to itself. Danezis et al. [10] developed a sealed-bid second-price auction to estimate the users' value of sensing data. Lee and Hoh [11] designed a reverse auction based dynamic incentive mechanism for purchasing users' sensing data. Jaimes et al. [14] further considered the users' location and a budget constraint. Yang et al. [13] designed an auction-based incentive mechanism, and proved this mechanism was computationally efficient, individually rational, profitable, and truthful. However, all of these studies failed to account for the online arrival of users.

To the best of our knowledge, there are few research work on the online mechanism design for crowdsourcing markets [24]–[26]. Singer et al. [24] and Singla et al. [25] presented pricing mechanisms for crowdsourcing markets based on the bidding model and the posted price model respectively. However, they focused only on a simple additive utility function instead of the submodular one. Badanidiyuru et al. [26] considered pricing mechanisms for maximizing the submodular utility function under the bidding model. However, they failed to consider the *consumer sovereignty* and the *time-truthfulness*.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we have investigated online incentive mechanisms for mobile crowdsourced sensing. We focus on the non-negative monotone submodular value function that can be applied in many realistic scenarios. We have designed two online mechanisms under different assumptions: *OMZ* can be applied to the zero arrival-departure interval case, and *OMG* can be applied to a more general case. We have proven that our mechanisms satisfy the *computational efficiency*, *individual rationality*, *budget feasibility*, *truthfulness*, *consumer sovereignty* and *constant competitiveness*.

The are several interesting open problems. Here, we aim at maximizing the value of services under a budget constraint. One future work is to design online mechanisms for minimizing payments while obtaining the specific value of services, where the value function is non-negative monotone submodular. Another interesting work is for maximizing the

profit of the crowdsourcer, namely the value of services minus the total cost of selected users, without budget constraint, where the profit function is non-monotone submodular, and can be negative.

REFERENCES

[1] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, and D. Zeinalipour-Yazti, "Crowdsourcing with smartphones," *IEEE Internet Computing*, pp. 36–44, 2012.
[2] "Sensorly," http://www.sensorly.com.
[3] E. Koukoumidis, L.-S. Peh, and M. R. Martonosi, "Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory," in *ACM MobiSys*, 2011, pp. 127–140.
[4] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *ACM SenSys*, 2008, pp. 323–336.
[5] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones," in *ACM SenSys*, 2009, pp. 85–98.
[6] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Earphone: an end-to-end participatory urban noise mapping system," in *ACM/IEEE IPSN*, 2010, pp. 105–116.
[7] M. Stevens and E. DHondt, "Crowdsourcing of pollution data using smartphones," in *ACM UbiComp*, 2010, pp. 1–4.
[8] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.
[9] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.
[10] G. Danezis, S. Lewis, and R. Anderson, "How much is location privacy worth," in *WEIS*, 2005.
[11] J. Lee and B. Hoh, "Sell your experiences: a market mechanism based incentive for participatory sensing," in *IEEE PerCom*, 2010, pp. 60–68.
[12] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand, "Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing," in *IEEE INFOCOM*, 2012, pp. 1701–1709.
[13] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *ACM MobiCom*, 2012.
[14] L. Jaimes, I. Vergara-Laurens, and M. Labrador, "A location-based incentive mechanism for participatory sensing systems with budget constraints," in *IEEE PerCom*, 2012, pp. 103–108.
[15] D. Zhao, X.-Y. Li, and H. Ma, "Omg: How much should i pay bob in truthful online mobile crowdsourced sensing?" Technical Report, 2013. [Online]. Available: http://arxiv.org/abs/1306.5677
[16] M. T. Hajiaghayi, R. Kleinberg, and D. C. Parkes, "Adaptive limited-supply online auctions," in *EC*, 2004, pp. 71–80.
[17] M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg, "Online auctions and generalized secretary problems," *ACM SIGecom Exchanges*, vol. 7, no. 2, pp. 1–11, 2008.
[18] M. Bateni, M. Hajiaghayi, and M. Zadimoghaddam, "Submodular secretary problem and extensions," in *APPROX-RANDOM*, 2010, pp. 39–52.
[19] A. V. Goldberg, J. D. Hartline, A. R. Karlin, M. Saks, and A. Wright, "Competitive auctions," *Games and Economic Behavior*, vol. 55, no. 2, pp. 242–269, 2006.
[20] Z. Bar-Yossef, K. Hildrum, and F. Wu, "Incentive-compatible online auctions for digital goods," in *SODA*, 2002, pp. 964–970.
[21] Y. Singer, "Budget feasible mechanisms," in *FOCS*, 2010.
[22] S. Khullera, A. Mossb, and J. Naor, "The budgeted maximum coverage problem," *Information Processing Letters*, vol. 70, pp. 39–45, 1999.
[23] X. Sheng, J. Tang, and W. Zhang, "Energy-efficient collaborative sensing with mobile phones," in *IEEE INFOCOM*, 2012, pp. 1916–1924.
[24] Y. Singer and M. Mittal, "Pricing mechanisms for crowdsourcing markets," in *WWW*, 2013, pp. 1157–1166.
[25] A. Singla and A. Krause, "Truthful incentives in crowdsourcing tasks using regret minimization mechanisms," in *WWW*, 2013, pp. 1167–1178.
[26] A. Badanidiyuru, R. Kleinberg, and Y. Singer, "Learning on a budget: posted price mechanisms for online procurement," in *EC*, 2012, pp. 128–145.

APPENDIX

A. Proof of Lemma 6

We consider two cases according to the total payment to the selected users at the last stage as follows.

Case (a): The total payment to the selected users at the last stage is at least αB, $\alpha \in (0, 1/2]$. In this case, since each selected user has marginal density at least ρ^*, so we have that

$$V(Z_2') \geq \rho^* \alpha B = \frac{\alpha \rho_1' B}{\delta} = \frac{2\alpha V(Z_1')}{\delta}.$$

Case (b): The total payment to the selected users at the last stage is less than αB, $\alpha \in (0, 1/2]$. There might be two reasons leading to that users from Z_2 are not selected in Z_2'. The first case is when the marginal densities of some users from Z_2 are less than ρ^*, and thus we do not select them. Even if these users are all in Z_2, their expected total payment is at most $B/2$. Because of submodularity, the expected total loss due to these missed users is at most

$$\rho^* \cdot \frac{B}{2} = \frac{\rho_1' B}{2\delta} = \frac{V(Z_1')}{\delta}.$$

The other case is when there is not enough budget to pay for some users whose marginal densities are not less than ρ^*. It means that the payment for such a user (for example, user i) is larger than $(1/2 - \alpha)B$, i.e., $V_i(\mathcal{S})/\rho^* > (1/2 - \alpha)B$; otherwise adding this user to Z_2' will not lead to that the total payment for Z_2' exceeds the stage-budget $B/2$. Because $\mathbb{E}[\rho_1'] \geq \rho$, we have that

$$\mathbb{E}[V_i(\mathcal{S})] > \mathbb{E}[\rho^*] \cdot (\frac{1}{2} - \alpha)B = \frac{(1 - 2\alpha)\mathbb{E}[\rho_1']B}{2\delta} \geq \frac{(1 - 2\alpha)\rho B}{2\delta}.$$

Because the expected total payment to all users in Z_2 is at most $B/2$, there cannot be more than $(\frac{\delta}{1 - 2\alpha} - 1)$ such users in Z_2. Since the value of each user is at most $V(Z)/\omega$, the expected total loss due to these missed users is at most $(\frac{\delta}{1 - 2\alpha} - 1)V(Z)/\omega$. Therefore, we have that

$$\begin{aligned}
\mathbb{E}[V(Z_2')] &\geq \mathbb{E}[V(Z_2)] - (\frac{\delta}{1 - 2\alpha} - 1)\frac{V(Z)}{\omega} - \frac{\mathbb{E}[V(Z_1')]}{\delta} \\
&\geq \frac{V(Z)}{2} - (\frac{\delta}{1 - 2\alpha} - 1)\frac{V(Z)}{\omega} - \frac{\mathbb{E}[V(Z_1')]}{\delta} \\
&\geq [\frac{1}{2} - (\frac{\delta}{1 - 2\alpha} - 1)\frac{1}{\omega} - \frac{1}{\delta}]\mathbb{E}[V(Z_1')].
\end{aligned}$$

Considering both of case (a) and (b), the ratio of $\mathbb{E}[V(Z_2')]$ to $\mathbb{E}[V(Z_1')]$ will be at least $2\alpha/\delta$, if it satisfies that

$$\frac{1}{2} - (\frac{\delta}{1 - 2\alpha} - 1)\frac{1}{\omega} - \frac{1}{\delta} = \frac{2\alpha}{\delta}. \tag{1}$$

Therefore, for a specific parameter ω, we can obtain the optimal ratio of $\mathbb{E}[V(Z_2')]$ to $\mathbb{E}[V(Z_1')]$ by solving the following optimization problem:

Maximize $\frac{2\alpha}{\delta}$ **subject to Eq. (1) and** $\alpha \in (0, 1/2]$.

When ω is sufficiently large (at least 12), we can obtain a constant ratio of $\mathbb{E}[V(Z_2')]$ to $\mathbb{E}[V(Z_1')]$. Specially, the optimal ratio approaches $1/4$ as $\omega \to \infty$ and $\delta \to 4$.