

## Article

# Truthful Incentive Mechanisms for Social Cost Minimization in Mobile Crowdsourcing Systems

Zhuojun Duan <sup>1</sup>, Mingyuan Yan <sup>2</sup>, Zhipeng Cai <sup>1,\*</sup>, Xiaoming Wang <sup>3</sup>, Meng Han <sup>1</sup> and Yingshu Li <sup>1</sup>

<sup>1</sup> Department of Computer Science, Georgia State University, Atlanta, GA 30302, USA; zduan2@student.gsu.edu (Z.D.); mhan7@student.gsu.edu (M.H.); yili@gsu.edu (Y.L.)

<sup>2</sup> Department of Computer Science and Information Systems, University of North Georgia, Dahlonega, GA 30597, USA; mingyuan.yan@ung.edu

<sup>3</sup> School of Computer Science, Shaanxi Normal University, Xi'an 710119, China; wangxm@snnu.edu.cn

\* Correspondence: zcai@gsu.edu; Tel.: +1-404-413-5721

Academic Editor: Yu Wang

Received: 10 February 2016; Accepted: 30 March 2016; Published: 6 April 2016

**Abstract:** With the emergence of new technologies, mobile devices are capable of undertaking computational and sensing tasks. A large number of users with these mobile devices promote the formation of the Mobile Crowdsourcing Systems (MCSs). Within a MCS, each mobile device can contribute to the crowdsourcing platform and get rewards from it. In order to achieve better performance, it is important to design a mechanism that can attract enough participants with mobile devices and then allocate the tasks among participants efficiently. In this paper, we are interested in the investigation of tasks allocation and price determination in MCSs. Two truthful auction mechanisms are proposed for different working patterns. A Vickrey–Clarke–Groves (VCG)-based auction mechanism is proposed to the continuous working pattern, and a suboptimal auction mechanism is introduced for the discontinuous working pattern. Further analysis shows that the proposed mechanisms have the properties of individual rationality and computational efficiencies. Experimental results suggest that both mechanisms guarantee all the mobile users bidding with their truthful values and the optimal maximal social cost can be achieved in the VCG-based auction mechanism.

**Keywords:** Mobile Crowdsourcing Systems (MCSs); incentive mechanism; cost minimization; auction

## 1. Introduction

Nowadays, new emerging embedded technology drives the rapid growth of mobile devices. With powerful processors, mobile devices such as smartphones, tablets, and watches can be used as portable computers to undertake heavy computational tasks. With the help of embedded sensors like Global Position System (GPS), accelerometers, and cameras, mobile devices can be used to sense and deliver information. On the other hand, the utilization of mobile devices is ubiquitous. Some works [1,2] show that almost 64% of adults own a smartphone and 42% of adults own a tablet in America as of October 2014. All of the above conditions, along with the mobility of users who carry these mobile devices and the convenient communication infrastructures enable mobile devices to connect to the Internet, stimulating the development of Mobile Crowdsourcing Systems (MCSs) [3]. The MCS is a new system model used to outsource tasks. Generally speaking, two types of participants exist in a MCS. One is the crowdsourcing platform, which acts as the server to publish tasks, determine the set of mobile devices to work on the tasks, and collect the final results. The other participants consist of users with mobile devices. They can participate to finish the tasks published by the crowdsourcing platform and get payments as rewards. Lots of tasks can be done by a MCS, such as information

collection, environmental monitoring, or customized survey. These tasks used to be performed by a specialist or an expert, but now through the MCS can be done by a group of undefined users with mobile devices.

A variety of MCS applications can be found in our daily life, among which applications focusing on the environment, infrastructure, and social activities are the three most popular categories. In the environmental MCS applications, such as Common Sense and Creek watch (introduced in [4,5], respectively), mobile devices can be used to monitor the environmental pollution levels. For example, microphones on mobile devices can monitor the noise information of a place and pictures can be taken by cameras to show the amount of trash in a park in Common Sense. Existing applications of the Infrastructure interested in the detection of traffic congestion, parking availability, and outages of public works. For example, mobile devices with CarTel [6] installed on cars can detect the speed and location of cars, and send the detected information to a data center. ParkNet [7] can help cars to find available parking places. Applications regarding social activities take advantage of users' willingness to share sensed information with each other [8,9]. All MCS applications require the participation of hundreds or thousands of mobile devices without deploying any static sensors or machines.

The enormous utilization potential of MCSs attracts lots of attention from researchers [10,11]. One of the most popular topics in MCSs is how to determine the best set of mobile devices to allocate the tasks (such as computational or sensing tasks) published by a MCS platform so that a predefined objective can be achieved. A commonly used objective is to optimize the social efficiency, such as maximizing social welfare or minimizing social cost. The fundamental of a MCS is to have enough participants. However, working for MCS platforms will consume users' resources, including execution capacity and battery. Joining a MCS will also put a threat on users' privacy. For example, the results submitted to MCS platform may expose users' locations. Considering the above-mentioned facts, some users may refuse to participate in the MCS. If the number of users with mobile devices is insufficient, the objective is impossible to be achieved. Thus, a MCS platform should provide enough reward for participants for incentive purposes.

In this paper, we focus on the design of an incentive mechanism for a MCS to minimize the social cost. The social cost represents the total cost of mobile devices when all tasks published by the MCS are finished. To achieve the objective in the MCS, we confront several challenges:

- *True cost revelation.* The cost of each mobile device for finishing a task is private. It is difficult to encourage all participants to report their real costs.
- *Minimal cost optimization.* Assume all users report their true costs to the MCS. Since mobile devices may vary in capacities and costs, it is hard to select the optimal set of users.
- *Incentive mechanism.* As discussed, the MCS platform should reward each user who works for it as incentives. Within the budget, the reward should be greater than the cost of the user. Deciding a proper reward for each participant is still challenging.

These challenges lead us to investigate an auction mechanism that concentrates on the trade between the MCS platform and mobile users. This paper begins with the assumption that the MCS platform publishes only one task in one round and the task consists of pieces of sub-tasks. Each user with a mobile device can work for one or more sub-tasks. The auction mechanism in our paper aims to minimize the social cost of mobile users while guaranteeing the truthful cost of bidding from each participating user.

Depending on the requirements of a MCS platform, there are two different working patterns. The first one is the *continuous* working pattern, which requires each participant to work on a set of continuous sub-tasks. We call another working pattern the *discontinuous* working pattern, where a participant can work for any set of sub-tasks. The detailed definition of the two working patterns are discussed in Section 4.

The main contributions of this paper are as follows:

1. The social cost minimization problem in a MCS has been discussed. We first present the working process of a MCS, and then build an auction market for the MCS where the MCS platform acts as an auctioneer and users with mobile devices act as bidders.
2. Depending on the different requirements of the MCS platform, we design a Vickrey-Clarke-Groves (VCG)-based auction mechanism for the continuous working pattern and a suboptimal auction mechanism for the discontinuous working pattern. Both of them can ensure that the bidding of users are processed in a truthful way and the utilities of users are maximized.
3. Experiments are conducted to verify performances of the proposed mechanisms. Results suggest that the two auction mechanisms achieve truthfulness and utility maximization. In addition, the VCG-based mechanism could guarantee the minimum social cost and the suboptimal mechanism is more computationally efficient.

The remainder of this paper is organized as follows: related works are shown in Section 2. In Section 3, we present the MCS model and analyze its working process. The auction problem is defined in Section 4, followed by two truthful auction mechanisms presented in Section 5. The experimental results and discussions are provided in Section 6. Conclusions and future works are shown in the last section.

## 2. Related Works

Recently, many commercial MCS applications have been released. These applications can be installed on mobile devices carried by users. After installation, these mobile users are able to undertake computational or sensing tasks. Then, all the results or information generated by the applications will be transmitted to a service center for final process. For example, GigWalk [12] can assist users in verifying the service quality and product placement. It can also provide reports about the graffiti at bus or train stations to the government. Additionally, GigWalk could work for real estate, consumer research, travel, advertising, and so on. Field Agent [13] is an application used for businesses. It can work for two tasks—audit and research. The audit task mainly focuses on information collection, which allows manufacturers and retailers to attract customers and spread information [14–16]. The research task is interested in gathering customers' feedback on products or services, so the businessmen can have a better insight of the market. In [4], the authors introduce a MCS application named Common Sense, which is used for pollution monitoring. Nericell [17] can be used to determine the average speeds or traffic delays, and DietSense [9] is proposed for health control. These applications suggest that the importance of MCS is growing in practical business fields.

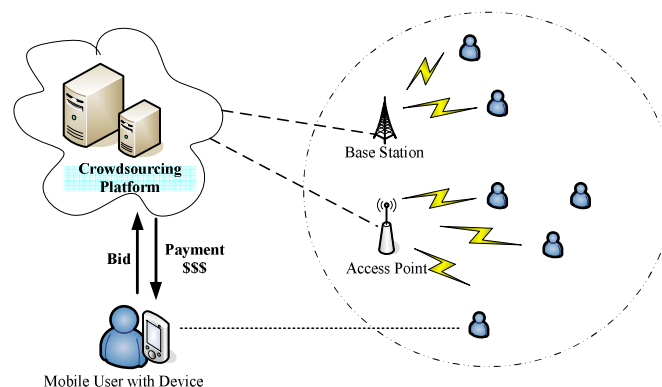
Many studies have been done on task allocation in MCSs. Most of them target the maximization of system efficiency. In [18], the authors design a fair energy-efficient allocation framework and propose two sensing task allocation algorithms: one is an offline allocation algorithm and the other is an online allocation algorithm. Ho and Vaughan [19] formalize the online task assignment problem, which makes the allocation decision upon arrival of each worker. Then, a two-phase exploration–exploitation assignment algorithm is proposed. Authors of [20] investigate the problem of task assignment and label inference for heterogeneous classification tasks. They derive a probably near-optimal adaptive assignment algorithm by applying online primal-dual techniques. An architectural model using the SLURM tool for efficient management in the MCS is outlined in [21]. The authors propose a novel idea of adaptive task scheduling which is based on the feedback of customer satisfaction. However, they don't consider the incentive mechanisms.

A handful of researchers put effort on the design of incentive mechanisms for the MCS. Yang *et al.* [22] consider two types of incentive mechanisms: platform-centric incentive mechanisms and user-centric incentive mechanisms. The first one is based on the Stackelberg game, in which the MCS platform has absolute control over the total budget to users, and users can only adjust their actions to meet the requirements of the platform. The roles of the platform and users are reversed in the user-centric incentive mechanisms. Each user reports the lowest price for selling a service to the MCS platform. In [23], the authors design a reward-based collaboration mechanism. The client publishes a

total reward to be shared among collaborators. The collaboration is successful when enough users are willing to collaborate. In order to attract more users to participate, [24] designs a novel Reverse Auction-based Dynamic Price (RADP) incentive mechanism. In this mechanism, users can sell their sensing data to a service provider by their claimed bid prices. Singla and Krause [25] exploit a link between procurement auctions and multi-armed bandits. Its mechanism design is budget feasible. In conclusion, most of the existing works concentrate on maximizing social efficiency and achieving fairness in MCSs.

### 3. System Model Overview

Figure 1 demonstrates an example of the Mobile Crowdsourcing System (MCS). The model includes two types of participants: a Crowdsourcing Platform (CP) and lots of Mobile Users with Devices (MUDs). The CP consists of several servers, which are deployed in the cloud and provide services for clients. A smartphone or a tablet carried by a user is regarded as a MUD. The CP communicates with MUDs via cellular networks or WiFi. The CP publishes a computational or a sensing task which contains a series of sub-tasks. Each sub-task only occupies a time slot. Any two time slots have no intersection. Each MUD is allowed to work on one or more time slots and provide computational or sensing services to the CP during these time slots. The concepts sub-task and time slot are used interchangeably in this paper. Working for computational or sensing tasks will bring battery consumption, computing capacity consumption, and privacy threats to MUDs. Thus, in order to stimulate MUDs' participation, the CP rewards these users who have been selected to provide services. For simplicity, we assume that the CP publishes one task in each round.



**Figure 1.** An overview of a Mobile Crowdsourcing System (MCS).

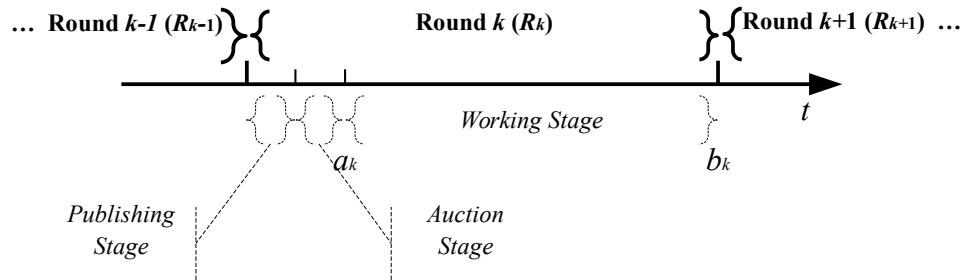
In general, as shown in Figure 2, the interactive process between the CP and MUDs have three stages in each round, including the publishing stage, auction stage, and working stage:

*Publishing stage.* In this stage, the CP decides the task that it plans to finish in this round. It generates the description of the task according to predefined functions and then publishes it among all participated MUDs.

*Auction stage.* After receiving the task description and requirements, each MUD decides its working plan. That is the subset of sub-tasks of task  $k$  it can work for. If a MUD can work for task  $k$ , it will continue to evaluate its cost. The MUD calculates its base price and submits a bid to the CP. The bid of a MUD consists of its working plan and the base price. After receiving the bids from MUDs, the CP will choose the winner set of MUDs, make the work schedule, determine their rewards and then announce the auction results to all participated MUDs.

*Working stage.* For each task  $k$ , its working stage starts at the start time  $a_k$  and ends at the end time  $d_k$ . During this stage, the CP will activate the MUDs in the winner set one by one based on its working schedule. Once activated, a MUD begins to work according to the requirements and then submits the result to the CP. The reward is given to the MUD once it finishes its claimed sub-tasks.

Different from the auction stage, the other three stages in the MCS are beyond the scope of this paper. We focus on designing efficient and effective auction mechanisms. Table 1 lists frequently used notations.



**Figure 2.** The interactive process between the crowdsourcing platform (CP) and mobile users with devices (MUDs).

**Table 1.** Table of Notations.

Notation	Description
CP	Crowdsourcing Platform
MUD	Mobile User Device
$V, v_i$	set of MUDs and MUD
$k$	round and task identifier
$R_k$	description of task $k$
$a_k, d_k$	the start time and end time of task $k$
$T_k$	set of sub-tasks in task $k$
$\tau_{ki}, \tau$	sub-task in task $k$ , sub-task
$\Omega_k, \Pi_k$	hardware parameters and working patterns
$b_i$	bid of MUD $v_i$
$\mathbf{b}_{-i}$	bids of all MUDs except $v_i$
$c_i, p_i, U_i$	cost, payment and utility of MUD $v_i$
$A_i$	asking price of MUD $v_i$
$S_i$	subset of sub-tasks MUD $v_i$ can work for
$W_k$	set of winner MUDs
$F$	number of MUDs

#### 4. Problem Formulation in Auction Stage

The working process of a MCS can be divided into infinite rounds with time. For any two rounds, their tasks are independent and the available MUDs are independent as well. Thus, we focus our investigation on the discussion of one round in detail. Assume there is a dense set of MUDs, represented as  $V_k = \{v_1, v_2, \dots, v_i, \dots, v_N\}$ , where  $|V_k| = N$ . At the beginning of round  $k$  ( $k$  is an integer denoting the identifier of one round), the CP publishes the task description  $R_k$ , defined as:

$$R_k = \{a_k, d_k, T_k, \Omega_k, \Pi_k\},$$

where  $a_k$  and  $d_k$  are the start and end time of task  $k$ , respectively.  $T_k = \{\tau_{k1}, \tau_{k2}, \dots, \tau_{kj}, \dots, \tau_{kM}\}$  represents the set of sub-tasks of task  $k$ . Each time slot represents a sub-task of task  $k$ .  $M$  is the size of  $T_k$  (the number of sub-tasks in task  $k$ ). The CP requires MUDs to work for task  $T_k$ . As shown in Figure 3, on the one hand, the durations of any two time slots  $\tau_{ki} \in T_k$  and  $\tau_{kj} \in T_k$ , where  $i \neq j$ , may vary from each other. On the other hand, all these time slots are distributed over the time line. The interval between two adjacent time slots could be larger than or equal to 0, but not smaller than 0, which means that no overlapping time interval exists between any two adjacent time slots.  $\Omega_k$  indicates the hardware requirements of task  $k$  on MUDs. Hardware requirements contain minimum computation speed, free storage capacities, and sensor types. It requires that only the MUDs

satisfying the requirements can bid for the sub-tasks at this round.  $\Pi_k$  indicates the requirement of task  $k$  regarding MUDs' working patterns. There are two kinds of working patterns: the continuous pattern ( $\Pi_k = C$ ) and the discontinuous pattern ( $\Pi_k = \bar{C}$ ).

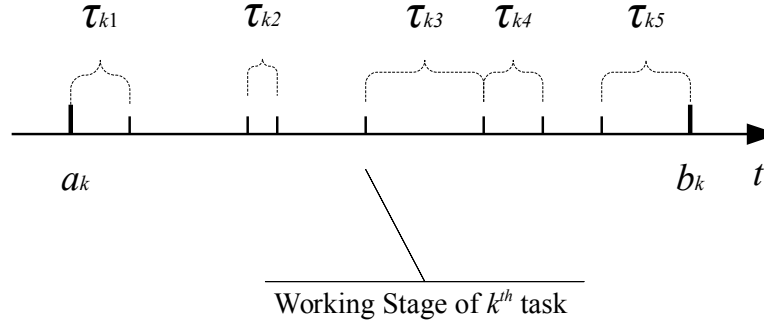


Figure 3. An example of sub-tasks in one round.

*Continuous case (C):*  $\forall v_i \in V_k$ , this working pattern requires the sub-tasks set  $S_i$  claimed by  $v_i$  are continuous ( $S_i$  represents the set of sub-tasks  $v_i$  can work for). That is,  $v_i$  is able to work continuously from the earliest sub-task to the last sub-task in  $S_i$ . For example, suppose five sub-tasks are included in task  $k$ , as shown in Figure 3.  $S_i = \{\tau_{k2}, \tau_{k3}, \tau_{k4}\}$  is an example that  $v_i$  works in a continuous working pattern, while  $S_i = \{\tau_{k2}, \tau_{k3}, \tau_{k5}\}$  is not.

*Discontinuous case ( $\bar{C}$ ):*  $\forall v_i \in V_k$ , in this case,  $v_i$  can work for any subset of  $T_k$ . For example, both  $S_i = \{\tau_{k2}, \tau_{k3}, \tau_{k4}\}$  and  $S_i = \{\tau_{k2}, \tau_{k3}, \tau_{k5}\}$  can be regarded as examples of discontinuous working patterns.

$\forall v_i \in V_k$ , after receiving a task description from the CP, if its hardware qualifies,  $v_i$  will decide the subset of sub-tasks, denoted as  $S_i \subseteq T_k$  according to the working patterns requirements of task  $k$ . Then,  $\forall v_i \in V_k$ , the bid of  $v_i$  can be represented as,

$$b_i = \{\Omega_i, \Pi_i, S_i, A_i\},$$

where  $A_i$  is  $v_i$ 's asking price when it works on the sub-tasks in  $S_i$  for the CP, which is also the base price. Because auction mechanisms are expected to be truthful, so  $A_i = c_i$ . In practice, the value of  $c_i$  can be estimated by  $v_i$ .  $A_i$  and  $c_i$  are used interchangeably in this paper. For convenience, all MUDs' costs are restricted to follow simple cost functions which makes all MUDs *single-minded* MUDs.

**Definition 1.** A cost function  $\mathbb{C}$  is called *single-minded* if there exists a set of sub-tasks  $S \subseteq T_k$  and a cost  $c$ ,  $\mathbb{C}(S^*) = c$  for all allocations  $S^* \subseteq S$  and  $\mathbb{C}(S^*) = \infty$  for all other  $S^*$ . A MUD bids with  $S$  and  $c$  is *single-minded*.

Definition 1 shows that the base price of each  $v_i$  will be same even though the set of sub-tasks  $S_i^*$  allocated to  $v_i$  is a subset of  $S_i$  in its bid after the auction stage. One step further, the MUD  $v_i$  wouldn't accept any allocation  $S^*$ , where  $\exists \tau \in T_k, \tau \in S^* \text{ but } \tau \notin S$ .

In the auction stage, the CP can be regarded as an auctioneer who makes the decision for the sub-tasks allocation and payment. MUDs act as bidders to make bids. We are interested in minimizing the social cost from a macroscopic and social perspective in this paper. The social cost is defined as the cost brought by the trading within the MCS. Formally, the objective can be written as,

$$\begin{aligned} \text{Minimize} \quad & \sum_{v_i \in W_k} (P_i + (c_i - p_i)) \\ \text{s.t.} \quad & W_k \subseteq V_k, T_k = \bigcup_{v_i \in W_k} S_i \end{aligned} \tag{1}$$



where  $W_k$  is the set of winner MUDs in this round.  $P_i$  is the price paid by the CP for using the computational or sensing services provided by  $v_i$ , which can be regarded as the cost of the CP.  $p_i$  is the payment winner  $v_i$  received from the CP when the assigned sub-tasks are done, and  $c_i$  is the cost of  $v_i$ . So, the social cost of winner  $v_i$  is  $c_i - p_i$ . For effectiveness, each sub-task  $\tau$  in  $T_k$  needs at least one MUD to work.

Based on the predefined model, we have  $P_i = p_i$  for each  $v_i \in W_k$ . Hence, the objective above can be rewritten as

$$\begin{aligned} & \text{Minimize} \quad \sum_{v_i \in W_k} c_i, \\ & \text{s.t.} \quad W_k \subseteq V_k, \quad T_k = \bigcup_{v_i \in W_k} S_i. \end{aligned} \quad (2)$$

Within the auction stage, the auctioneer CP should design a proper auction mechanism with efficient sub-tasks allocation and rewards determination to minimize the social cost. Under the auction mechanism chosen by the CP, each participated MUD bids with a strategy which maximizes its utility. The utility of  $v_i$  (denoted by  $U_i$ ) in one round can be defined as:

$$U_i = \begin{cases} p_i - c_i & v_i \text{ wins,} \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

In order to achieve the objective in an efficient and effective way, the auction mechanisms used by the CP should have the following properties:

**Individual Rationality.** All MUDs are self-interested to benefit themselves. Thus, the utility of any MUD in each round should be non-negative:  $U_i \geq 0$ .

**Truthfulness.** The mechanisms are considered truthful when the four values ( $\Omega_i$ ,  $\Pi_i$ ,  $S_i$ ,  $A_i$ ) in the bid of each MUD are truthful. The utility of  $v_i$  will be maximized when it bids truthfully and  $v_i$  cannot improve its utility through any misreport,

$$U_i(b_i, \mathbf{b}_{-i}) \geq U_i(\hat{b}_i, \mathbf{b}_{-i}), \quad (4)$$

where  $\mathbf{b}_{-i} = \{b_1, \dots, b_{i-1}, b_{i+1}, \dots, b_n\}$  represents the set of truthful bids of all MUDs excluding  $v_i$ .  $b_i$  is the truthful bid of  $v_i$ , and  $\hat{b}_i \neq b_i$ . If an auction mechanism satisfies this property, *Nash Equilibrium* exists [26]. Misreports of the first two values (hardware parameters and working pattern of a MUD) in a bid can be easily detected by the CP through the submitted results from MUDs. Thus, the truthfulness of the first two values is guaranteed. We focus on the truthfulness of the last two values in a bid: claimed set of sub-tasks and asking price.

**Computational Efficiency.** An auction mechanism is considered computationally efficient if the task allocation and payment decision can be made in polynomial time.

Only when the above three properties are satisfied at the same time can an auction mechanism be regarded as useful. Without individual rationality, a MUD may receive negative utility and refuse to participate in the MCS. Then, because the  $c_i$  in bid  $b_i$  is private to  $v_i$ , the CP wouldn't know it. If an auction mechanism is truthful, all MUDs only need to bid with their true costs:  $A_i = c_i$ , which not only simplifies the strategies, but also avoids manipulation. Finally, computational efficiency will guarantee that the auction mechanism can be practically implemented.

## 5. Design of Incentive Auction Mechanisms

Formally, an auction mechanism contains two phases: winner MUDs set the selection and payment decision. Specifically, the most challenging and important part of the auction mechanism design is truthfulness. According to the characterization of truthful auction mechanism concluded in [26], we have:

**Theorem 2.** For any fixed bids  $\mathbf{b}_{-i}$ , an auction mechanism is truthful to MUD  $v_i$  if and only if the winner MUDs set selection algorithm is monotone and the payment for each winner MUD is critical.

**Definition 3.** If the MUD  $v_i$  is selected as a winner when it bids with  $S_i^*$  and  $A_i^*$ , and  $v_i$  will still be selected for any  $(S'_i, A'_i)$ , where  $S'_i \supseteq S_i^*$  and any bid with  $A'_i \leq A_i^*$ . The process in the selection of winner MUDs set is monotone.

**Definition 4.** Critical payment. There exists a critical payment  $c_i^c$  for each winner MUD  $v_i$ , which is independent to the asking price  $a_i$  in its bid.  $v_i$  will win when it bids with any  $(S_i, A'_i)$ , where  $A'_i \leq c_i^c$ . Otherwise,  $v_i$  loses.

### 5.1. Mechanism with Optimal Social Cost

Based on Section 4, we can tell that different tasks may have different working pattern requirements. If a task  $k$  requires MUDs to work in the continuous pattern ( $\Pi_k = C$ ), the optimal solution to the minimization Problem (2) can be achieved through dynamic programming. Thus, in this case, a well known VCG-based auction mechanism is a good choice. The detailed design is as follows:

*Step 1,* Given the bids of all candidate MUDs, use the dynamic programming method, shown in Algorithm 1, to compute the optimal winner set  $W$ . In Algorithm 1, Line (8) represents the optimal substructure, recording the minimal cost for sub-task  $\tau$  in  $T$  if adding the  $v_n$  to the winner set  $W$ . Notation  $\tau.MUD$  is used to indicate that the winner MUD will work for the sub-task  $\tau$ . This algorithm returns three results: the winner set  $W$ , the work schedule, and minimal social cost of MUDs in  $W$ .

The complexity of Algorithm 1 is  $O(mn)$ , which indicates that the VCG-based auction is practical.

*Step 2,* Calculate the payment  $p_i$  of each  $v_i$  in the winner set  $W$ . The payment for each winner  $v_i$  is defined as the increase in the total social cost brought by its contribution, as

$$p_i = \sum_{v_j \in W_{-i}} c_j - \sum_{v_j \in W, v_j \neq v_i} c_j, \quad (5)$$

where  $W_{-i}$  is the obtained winner set without  $v_i$ 's participation.

The truthfulness and individual rationality of VCG-based mechanisms have been proven in [26].

### 5.2. Mechanism with Suboptimal Social Cost

If the working pattern is discontinuous, the winner set determination Problem (2) can be regarded as a classical set cover problem. Because the set cover problem has been proven to be NP-hard, it is impossible to obtain the optimal solution in a MCS with large scale. Without optimal winner selection, truthfulness cannot be guaranteed by VCG-based auction mechanism. Therefore, we propose another mechanism with suboptimal social cost, acceptable computational complexity, and truthfulness.

The detailed winner set determination algorithm is shown in Algorithm 2. It consists of two steps. First, sort all MUDs in ascending order, according to their average cost  $\frac{c_i}{|S_i|}$  (as shown in line (2)). Then, MUDs will be added to winner set  $W$  one by one, according to the ascending order derived in the last step, until all sub-tasks in  $T$  are covered, as shown in lines (4–13). The payment of each winner  $v_i$  is determined based on Algorithm 3. Specifically, it first reorders all MUDs, excluding  $v_i$ , in ascending order based on their average cost  $\frac{c_j}{|S_j|}$  (as shown in line (2)). Then find the least position  $j$  in the order that:  $v_i$  may lose in the case of  $\frac{c_i}{|S_i|} > \frac{c_j}{|S_j|}$ .



**Algorithm 1** Optimal Winner MUDs Set Selection**Input:**

Set of sub-tasks  $T$ ,  $|T| = M$ .

$(S_1, c_1), \dots, (S_i, c_i), \dots, (S_N, c_N)$ ,

Working pattern: continuous.

**Output:**

miniCost, Winner set  $W$ ;

Working schedule  $\tau.MUD, \forall \tau \in T$ .

```

1: Initialization:
2: for each  $\tau$  in  $T$  do
3:    $\tau_{cost} = \text{MAX.VALUE}$ ;
4: end for
5: for each  $\tau$  in  $T$  do
6:   for each  $v_i$  in  $V$  do
7:     if  $\tau$  is in  $S_i$  then
8:        $currentMini = (\tau_0 == \tau_i = S_i.first)? c_i : (c_i + \tau_{i-1}.cost)$ ;
9:     end if
10:    if  $currentMini < \tau.cost$  then
11:       $\tau.cost = currentMini$ ;
12:       $\tau.MUD = v_i$ ;
13:      for each  $\tau'$  in  $S_n$  do
14:         $\tau'.cost = currentMini$ ;
15:         $\tau'.MUD = v_i$ ;
16:      end for
17:    end if
18:  end for
19: end for
20: for each  $\tau$  in  $T$  do
21:   Put  $\tau.MUD$  into  $W$ ;
22: end for
23:  $miniCost = T.last.cost$ ;

```

**Lemma 5.** The winner MUDs set selection provided in Algorithm 2 is monotonic.

**Proof.** Suppose  $v_i$  is selected as a winner by bidding with  $c_i$  and  $S_i$ , its average cost is  $\alpha_i = \frac{c_i}{|S_i|}$ . Let  $c'_i \leq c_i$ , and  $S_i$  remain unchanged, we should prove that  $v_i$  would still win when bidding with  $c'_i$  and  $S_i$ . The new average cost is  $\alpha'_i = \frac{c'_i}{|S_i|}$ . Since  $\alpha'_i \leq \alpha_i$ ,  $v_i$  will be selected earlier by Algorithm 2. It is easy to know that  $v_i$  will continue to win with bidding with  $c_i$  and  $S'_i$ , where  $S'_i \supseteq S_i$ . For the reason that the new average cost  $\alpha'_i = \frac{c'_i}{|S'_i|}$  is smaller than  $\alpha_i = \frac{c_i}{|S_i|}$ .  $\square$

**Lemma 6.** The payment  $p_i$  to each  $v_i \in W$  is equal to its critical cost  $c_i^c$ .

**Proof.** Assume that critical cost of each MUD is equal to its payment, that is,  $\forall v_i \in W, c_i^c = p_i$ . Based on Algorithm 3,  $c_i^c = p_i = \frac{c_j}{|S_j|} |S_i|$ , where  $S_i \subset \bigcup_{i'=1, \dots, j} S_{i'}$ . If  $v_i$  bids with  $c'_i > c_i^c$ , then  $\frac{c'_i}{|S_i|} > \frac{c_j}{|S_j|}$ , indicating that the average cost of  $v_i$  is larger than the average cost of  $v_j$ , the position of  $v_j$  in this new order is before  $v_i$ . In this case, once the  $v_j$  is in  $W$ ,  $v_i$  will have no chance to be selected as a winner. On the other hand, if  $v_i$  bids with  $c'_i \leq c_i^c$ , it will still be a winner according to Lemma 5. Thus the assumption is verified.  $\square$

**Theorem 7.** The suboptimal cost mechanism is truthful.

**Algorithm 2** Suboptimal Winner Set Selection**Input:**Set of sub-tasks  $T$ . $(S_1, c_1), \dots, (S_i, c_i), \dots, (S_N, c_N),$ **Output:** $MiniCost$ , Winner set  $W$ ;Working schedule  $\tau.MUD, \forall \tau \in T$ .1: *Initialization*:2:  $\forall v_i \in V$ , sort ascendingly:

$$\frac{c_1}{|S_1|} \leq \frac{c_2}{|S_2|} \leq \dots \leq \frac{c_N}{|S_N|},$$

3:  $U = \emptyset, j=1$ ;4: **while**  $U \neq T$  **do**5:   **if**  $U \cap S_j \neq S_j$  **then**6:     put  $v_j$  into  $W, U = U \cup S_j$ ,7:      $miniCost = miniCost + c_j$ ;8:   **end if**9:   **for** Each  $\tau$  in  $S_j$  **do**10:      $\tau.MUD = v_j$ ;11:   **end for**12:    $j++$ ;13: **end while****Algorithm 3** Price Determination**Input:**Set of sub-tasks  $T$ , $(S_1, c_1), \dots, (S_i, c_i), \dots, (S_N, c_N),$ Winner set  $W$ ;**Output:**Price  $p_i, \forall v_i \in W$ ;1: **for** each  $v_i$  in  $W$  **do**2:    $\forall v_j \in V \setminus v_i$ , sort ascendingly:

$$\frac{c_1}{|S_1|} \leq \frac{c_2}{|S_2|} \leq \dots \leq \frac{c_{N-1}}{|S_{N-1}|};$$

3:   Find the least index  $j$  that  $S_i \subset \bigcup_{i'=1, \dots, j} S_{i'}$ ;4:   The payment for MUD  $v_i$  is:

$$p_i = \frac{c_j}{|S_j|} |S_i|;$$

5: **end for****Proof.** With Lemmas 5 and 6, this theorem can be proven based on Theorem 2.  $\square$ **Theorem 8.** The suboptimal cost mechanism is individual rational.**Proof.** For  $\forall v_i \in V_k$ , its payment  $p_i$  is equal to its critical cost  $c_i^c$ . If  $v_i$  wins, there must be  $c_i \leq c_i^c$ . Hence,  $U_i = p_i - c_i \geq 0$ . Or  $v_i$  loses, its utility is 0. Individual rationality is guaranteed.  $\square$ **Theorem 9.** The suboptimal cost mechanism is computationally efficient.

**Proof.** The complexity of Algorithm 2 is  $O(n^2)$ . The complexity of Algorithm 3 is  $O(n^2)$ . The suboptimal social cost auction mechanism can be implemented with an acceptable time complexity.  $\square$

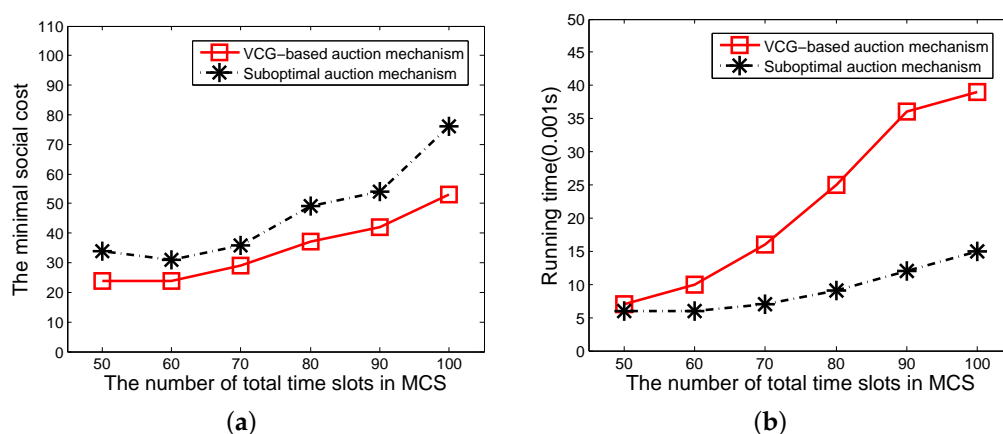
## 6. Performance Evaluation

To evaluate the performance of the incentive auction mechanisms proposed in this paper, experiments are conducted.

### 6.1. Continuous Working Pattern

We set the number of sub-tasks in one round to vary from 50 to 100. The number of MUDs is fixed as 30 ( $F = 30$ ). Each selects a subset of continuous sub-tasks and the size of the subset is randomly chosen from [3, 15]. The cost of each MUD is distributed over [10, 15] uniformly. The VCG-based auction mechanism is designed for the continuous working pattern. Because the continuous case can be regarded as a special case of discontinuous working pattern, the suboptimal auction mechanism can also be used to solve the problem here. Both the VCG-based auction mechanism and suboptimal auction mechanism are evaluated in this section.

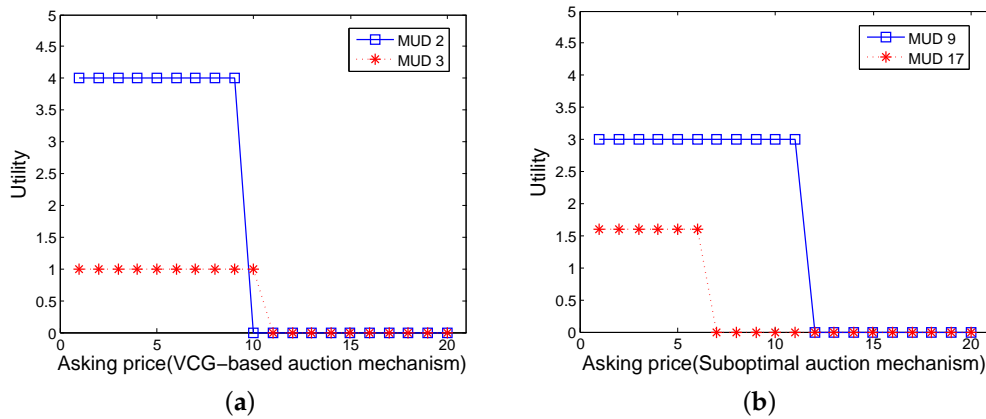
In this experiment, the number of sub-tasks changes from 50 to 100, the remaining parameters are set as above. We first compare the performances of the VCG-based auction mechanism and the suboptimal auction mechanism regarding the social cost and running time. The results of the social costs of the two auction mechanisms are shown in Figure 4a. The social cost of the VCG-based mechanism is smaller than the suboptimal auction mechanism because the dynamic winner set selection algorithm used in the VCG-based mechanism can find the subset of MUDs which is able to minimize the social cost. However, when considering the running times, as shown in Figure 4b, the suboptimal auction mechanism outperforms the VCG-based auction mechanism. We can also observe that with the increase of the number of total sub-tasks, both the cost and running time increase. When the number of sub-tasks increases, the running time of the VCG-based auction mechanism and the suboptimal auction mechanism increase by 700% and 200%, respectively. However, the difference of social costs between two mechanisms keeps steady at the same time. Thus, it is better to use the VCG-based auction mechanism in the continuous working pattern on the condition that a task has fewer sub-tasks.



**Figure 4.** (a) The social cost of two auction mechanisms in continuous working pattern; (b) The running time of two auction mechanisms in continuous working pattern. VCG: Vickrey–Clarke–Groves.

Then we try to observe the utilities of winner MUDs by the VCG-based auction mechanism. For simplicity, we choose two MUDs randomly, denoted as MUD 2 and MUD 3. We allow the two MUDs to ask different prices and show the truthfulness in Figure 5a. It is shown that both the

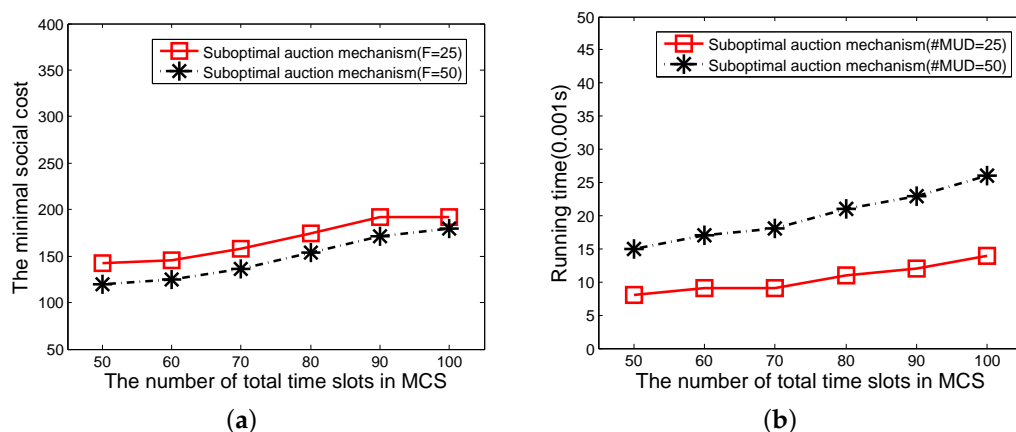
MUD 2 and MUD 3 will reach their maximal utilities when they ask the prices truthfully:  $A_2 = c_2 = 10$  and  $A_3 = c_3 = 9$ . Figure 5b presents the utilities of MUD 9 and MUD 17 in the suboptimal auction mechanism. Note that MUD 9 and MUD 17 are chosen randomly too. Similarly, they will also achieve their maximal utilities when acting truthfully:  $A_9 = c_9 = 9$  and  $A_{17} = c_{17} = 5$ .



**Figure 5.** (a) The utility of MUD 2 and MUD 3 by VCG-based auction mechanism in continuous working case; (b) The utility of MUD 9 and MUD 17 by suboptimal auction mechanism in continuous working case.

## 6.2. Discontinuous Working Pattern

The experimental setting of the discontinuous working pattern is similar to the continuous working pattern. Besides, each MUD could select the subset of sub-tasks randomly. Figure 6a,b shows the results of suboptimal auction mechanism of the social cost and running time when the number of MUDs is 30 and 50 ( $F = 25$  and  $F = 50$ ), respectively. Because the more sub-tasks that are contained within a task, the more works need to be done. We can see that when the number of sub-tasks of a task changes from 50 to 100, both the social cost and the running time increase (see Figure 6a,b). So, it is a trade-off between objective and efficiency in the suboptimal auction mechanism.



**Figure 6.** (a) The social cost of suboptimal auction mechanism in discontinuous working pattern; (b) The running time of suboptimal auction mechanism in discontinuous working pattern.

Then we try to verify the truthfulness of the suboptimal auction mechanism in the discontinuous working pattern. MUD 16 and MUD 19 are picked randomly, where  $c_{16} = 14$  and  $c_{19} = 10$ . Let the

two MUDs ask different prices from their true costs, their utilities are shown in Figure 7. We can see that both MUDs will get their maximal utilities when asking prices truthfully.

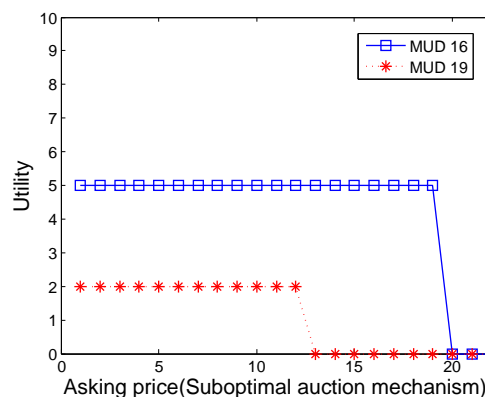


Figure 7. The utility of MUD 16 and MUD 19 by suboptimal auction mechanism in discontinuous case.

## 7. Conclusions

In this paper, we investigate the incentive auction mechanisms for mobile crowdsourcing systems. We consider two working patterns in works allocation: the continuous working pattern and the discontinuous working pattern. The objective of the MCS platform is to minimize the social cost in both cases. To achieve the truthfulness, individual rationality, and computational efficiency, we design a VCG-based auction mechanism for the continuous case and a suboptimal auction mechanism for the discontinuous case. We have proven that the two mechanisms can implement the three properties simultaneously. In the future, we plan to design an online incentive mechanism to minimize the social cost and try to maximize the utility of each participated MUD.

**Acknowledgments:** This work is partly supported by the NSF under grant No. CNS-1252292, and the NSFC under grants No. 61373083, 61370084 and 61502116.

**Author Contributions:** The work presented in this paper is a collaborative development by all of the authors. Zhipeng Cai and Zhuojun Duan contributed to the idea of the incentive mechanisms and designed the algorithms. Mingyuan Yan, Meng Han, Xiaoming Wang and Yingshu Li develop and analyze the experimental results. All the authors involve in writing the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Mobile Technology Fact Sheet. Available online: <http://www.pewinternet.org/fact-sheets/mobile-technology-fact-sheet/> (accessed on 3 November 2015).
2. Chatzimilioudis, G.; Andreas, K.; Christos, L.; Demetrios, Z.-Y. Crowdsourcing with smartphones. *IEEE Internet Comput.* **2012**, *5*, 36–44.
3. Fuchs-Kittowski, F.; Daniel, F. Architecture of mobile crowdsourcing systems. In *Collaboration and Technology*; Springer International Publishing: Santiago, Chile, 2014; pp. 121–136.
4. Dutta, P.; Aoki, P.M.; Kumar, N.; Mainwaring, A.; Myers, C.; Willett, W.; Woodruff, A. Common sense: Participatory urban sensing using a network of handheld air quality monitors. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, Berkeley, CA, USA, 4–6 November 2009; ACM: Berkeley, CA, USA, 2009.
5. Kim, S.; Robson, C.; Zimmerman, T.; Pierce, J.; Haber, E.M. Creek watch: Pairing usefulness and usability for successful citizen science. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vancouver, BC, Canada, 7–12 May 2011.
6. Hull, B.; Bychkovsky, V.; Zhang, Y.; Chen, K.; Goraczko, M.; Miu, A.; Shih, E.; Balakrishnan, H.; Madden, S. CarTel: A distributed mobile sensor computing system. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, Boulder, CO, USA, 1–3 November 2006.

7. Mathur, S.; Jin, T.; Kasturirangan, N.; Chandrasekaran, J.; Xue, W.; Gruteser, M.; Trappe, W. Parknet: Drive-by sensing of road-side parking statistics. In Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, San Francisco, CA, USA, 15–18 June 2010.
8. Eisenman, S.B.; Miluzzo, E.; Lane, N.D.; Peterson, R.A.; Ahn, G.S.; Campbell, A.T. BikeNet: A mobile sensing system for cyclist experience mapping. *ACM Trans. Sens. Netw.* **2009**, *6*, 6.
9. Reddy, S.; Parker, A.; Hyman, J.; Burke, J.; Estrin, D.; Hansen, M. Image browsing, processing, and clustering for participatory sensing: Lessons from a DietSense prototype. In Proceedings of the 4th Workshop on Embedded Networked Sensors, Cork, Ireland, 25–26 June 2007.
10. Li, J.; Cai, Z.; Yan, M.; Li, Y. Using crowdsourced data in location-based social networks to explore influence maximization. In Proceedings of the 35th Annual IEEE International Conference on Computer Communications, INFOCOM, San Francisco, CA, USA, 10–15 April 2016.
11. Wang, Y.; Cai, Z.; Ying, G.; Dong, Y. An incentive mechanism with privacy protection in mobile crowdsourcing systems. *Comput. Netw.* **2016**, doi:10.1016/j.comnet.2016.03.016. (In Press)
12. Gigwalk. Available online: <http://www.gigwalk.com/> (accessed on 2 November 2016).
13. Field Agent. Available online: <https://fieldagent.net/> (accessed on 2 November 2016).
14. Han, M.; Yan, Y.; Cai, Z.; Li, Y. An exploration of broader influence maximization in timeliness networks with opportunistic selection. *J. Netw. Comput. Appl.* **2016**, *63*, pp. 39–49.
15. Wang, X.; Lin, Y.; Zhao, Y.; Zhang, L.; Liang, Y.; Cai, Z. A novel approach for inhibiting misinformation propagation in human mobile opportunistic networks. *P2P Netw. Appl.* **2016**, doi:10.1007/s12083-016-0438-3.
16. He, Z.; Cai, Z.; Wang, X. Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks. In Proceedings of the 2015 IEEE 35th International Conference on Distributed Computing Systems (ICDCS), Columbus, OH, USA, 29 June–2 July 2015.
17. Mohan, P.; Padmanabhan, V.N.; Ramjee, R. Nericell: Rich monitoring of road and traffic conditions using mobile smartphones. In Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, Raleigh, NC, USA, 4–7 November 2008.
18. Zhao, Q.; Zhu, Y.; Zhu, H.; Cao, J.; Xue, G.; Li, B. Fair energy-efficient sensing task allocation in participatory sensing with smartphones. In Proceedings of the 2014 IEEE INFOCOM, Toronto, ON, Canada, 27 April–2 May 2014.
19. Ho, C.J.; Vaughan, J.W. Online task assignment in crowdsourcing markets. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence Toronto, Toronto, ON, Canada, 22–26 July 2012; Volume 12.
20. Ho, C.J.; Jabbari, S.; Vaughan, J.W. Adaptive task assignment for crowdsourced classification. In Proceedings of the 30th International Conference on Machine Learning (ICML-13), Atlanta, GA, USA, 16–21 June 2013.
21. Nunia, V.; Kakadiya, B.; Hota, C.; Rajarajan, M. Adaptive task scheduling in service oriented crowd using slurm. In *Distributed Computing and Internet Technology*; Springer: Berlin, Germany; Heidelberg, Germany, 2013; pp. 373–385.
22. Yang, D.; Xue, G.; Fang, X.; Tang, J. Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing. In Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Istanbul, Turkey, 22–26 August 2012.
23. Duan, L.; Kubo, T.; Sugiyama, K.; Huang, J.; Hasegawa, T.; Walrand, J. Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing. In Proceedings of the 2012 IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012.
24. Lee, J.S.; Hoh, B. Sell your experiences: A market mechanism based incentive for participatory sensing. In Proceedings of the 2010 IEEE International Conference on Pervasive Computing and Communications (PerCom), Mannheim, Germany, 29 March–2 April 2010.
25. Singla, A.; Krause, A. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; International World Wide Web Conferences Steering Committee: Rio de Janeiro, Brazil, 2013.
26. Vazirani, V.V.; Roughgarden, T.; Tardos, E. (Eds.) *Algorithmic Game Theory*; Cambridge University Press: Cambridge, UK, 2007; Volume 3, doi:10.1017/CBO9780511800481.

