

# Truthful Scheduling Mechanisms for Powering Mobile Crowdsensing

Kai Han, Chi Zhang, Jun Luo, Menglan Hu and Bharadwaj Veeravalli

**Abstract**—Mobile crowdsensing leverages mobile devices (e.g., smart phones) and human mobility for pervasive information exploration and collection; it has been deemed as a promising paradigm that will revolutionize various research and application domains. Unfortunately, the practicality of mobile crowdsensing can be crippled due to the lack of incentive mechanisms that stimulate human participation. In this paper, we study incentive mechanisms for a novel Mobile Crowdsensing Scheduling (MCS) problem, where a mobile crowdsensing application *owner* announces a set of sensing tasks, then human *users* (carrying mobile devices) compete for the tasks based on their respective sensing costs and available time periods, and finally the owner schedules as well as pays the users to maximize its own sensing revenue under a certain budget. We prove that the MCS problem is NP-hard and propose polynomial-time approximation mechanisms for it. We also show that our approximation mechanisms (including both offline and online versions) achieve desirable game-theoretic properties, namely truthfulness and individual rationality, as well as  $\mathcal{O}(1)$  performance ratios. Finally, we conduct extensive simulations to demonstrate the correctness and effectiveness of our approach.



## 1 INTRODUCTION

With the proliferation of palm-size mobile devices (smart phones, PDAs, etc.), we have a new tool for pervasive information collection, sharing, and exploration. For those information that traditionally require specific (possibly very expensive) instruments or devices to gather, they can now be outsourced to human crowds. Moreover, as this tool relies on human mobility and activity to bring their mobile devices around, it also introduces a new type of social action: mobile crowdsensing. Recently, there have emerged numerous systems based on this idea across a wide variety of research and application domains, such as healthcare, social networks, safety, environmental monitoring, and transportation [1], [2].

Whereas mobile crowdsensing appears to be a promising paradigm that will revolutionize many research and application domains and ultimately impact on our everyday life significantly, it cannot take place spontaneously in practice, as many other social actions. Apparently, participating in a mobile crowdsensing task usually requires a mobile device carrier (*users* hereafter) to move to specific areas where data gathering is required, to turn on his/her sensors for gathering data (e.g., GPS locations), and to upload the sensing data to an online server. These actions inevitably incur sensing costs in terms of, for example, (device) energy consumption/depreciation

and Internet access. Therefore, from a pragmatic point of view, human crowds may not be willing to participate in mobile crowdsensing unless they are incentivized. Therefore, proper incentive mechanisms are crucial for enabling mobile crowdsensing, and this intriguing problem has started to attract attentions very recently [3]–[7].

Designing incentive mechanisms for mobile crowdsensing is challenging, particularly because the designer face rational but selfish users who can act **strategically** (i.e., lying about their private information) to maximize their own utilities. To handle this issue, a mechanism needs to motivate the users to report their real private information, or in game theoretical term, the mechanism should be *truthful* [8]. Certain existing proposals [3], [4], [7] do not take truthfulness into account for the designed incentive mechanisms. Such mechanisms, though being able to motivate user participation in a mobile crowdsensing application, may end up costing the application *owner* a big fortune to obtain a certain sensing revenue.

Another important aspect in designing effective incentive mechanisms for mobile crowdsensing is that proper spatial/temporal requirements should be well satisfied to maximize the revenue gained from crowdsensing. This is due to the observation that, in many crowdsensing scenarios (e.g., environmental/traffic monitoring), the users must sense at specific locations for sufficient time to contribute meaningful data [7], [9].

Recently, the truthfulness problem in mobile crowdsensing has attracted great interests and some related proposals have also noticed the spatial coverage problems in crowdsensing (e.g., [4], [10]). However, the temporal aspect of mobile crowdsensing has been less investigated or even ignored by the literature. Most of the existing work assume that the sensing tasks can be completed instantly and the sensing costs/revenue of the users are irrelevant of the time period in which the

- Kai Han is with School of Computer Science and Technology / Suzhou Institute for Advanced Study, University of Science and Technology of China, China.  
E-mail: hankai@ustc.edu.cn
- Chi Zhang, Jun Luo and Menglan Hu are with School of Computer Engineering, Nanyang Technological University, Singapore.  
E-mail: {czhang8, junluo, mlhu}@ntu.edu.sg
- Bharadwaj Veeravalli is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore.  
E-mail: elebv@nus.edu.sg

users sensed (e.g., [10], [11]). Nevertheless, as sensing for longer time will inevitably cost more users' resources but contribute more sensing data, the sensing costs/revenue of the users may be greatly affected by the sensing time in many scenarios.

In this proposal, we look into the temporal features of mobile crowdsensing and investigate a novel time-scheduling problem, where a crowdsensing owner announces a set of sensing tasks with respective values, and users with different available time and sensing costs bid for these tasks. As the sensing revenue/cost is significantly affected by the sensing time, we design intelligent scheduling mechanisms to *simultaneously* allocate time slots and sensing tasks to users, hence maximize the total sensing revenue under a limited budget. Compared with existing work, our mechanism design problem is even more challenging since i) it faces multi-parameter environments where users' private information is multi-dimensional, ii) it has to schedule users properly for revenue maximization, and iii) it should be able to handle the dynamic arrivals of the users. To the best of our knowledge, these issues have never been tackled holistically in the literature by far.

Our mechanisms are designed based on game-theoretic methods and achieve multiple performance objectives including truthfulness, individual rationality [8], provable approximation ratios, as well as computational efficiency. Moreover, the proposed mechanisms work for both offline (users all arrive together) and online (user may arrive sequentially) cases. In summary, we have the following major contributions in this paper:

- We formally formulate the Mobile Crowdsensing Scheduling (MCS) problem and prove that it is NP-hard.
- We propose an offline polynomial-time mechanism for the MCS problem with  $\mathcal{O}(1)$  approximation ratio, which is truthful given that the users strategically report their multi-dimensional private information including the sensing costs and available time periods.
- We propose an online polynomial-time mechanism for the MCS problem with  $\mathcal{O}(1)$  competitive ratio, which is truthful if the users strategically report their sensing costs.
- We conduct extensive simulations, and the simulation results demonstrate the effectiveness of our approach.

The remaining of our paper is organized as follows. We introduce the models and assumptions in Sec. 2, where we also formulate the MCS problem. Then we first present an approximation algorithm for the MCS problem in Sec. 3. Based on this algorithm, we further propose truthful mechanisms for the MCS problem under both offline and online settings in Sec. 4 and Sec. 5, respectively. We report the results of our extensive simulations in Sec. 6. We finally discuss the related work in Sec. 7, as well as conclude our paper in Sec. 8. In

order to maintain fluency, we only prove a few crucial theorems in the main texts but postpone most of the (sketched) proofs to the Appendix.

## 2 MODELING AND PROBLEM FORMULATION

We formally introduce the assumptions and definitions for the MCS problem in this section.

### 2.1 The Mobile Crowdsensing Scheduling Problem

We assume that a mobile crowdsensing application *owner* announces a set of *sensing tasks*  $\mathcal{K} = \{K_1, K_2, \dots, K_m\}$ , and that performing any task  $K_i : 1 \leq i \leq m$  per unit time has a *sensing value*  $u_i \in \mathbb{R}^+$  to the owner, whereas performing any task for less than one unit time has a sensing value 0. We also assume that the owner holds a *budget*  $G \in \mathbb{R}^+$ : the maximum amount of total payment that it is willing to make for outsourcing the sensing tasks in  $\mathcal{K}$  to others.

Suppose that a set of *users* (or sensor carriers), denoted by  $\mathcal{A} = \{A_1, A_2, \dots, A_n\}$ , may potentially perform the sensing tasks in  $\mathcal{K}$ . For simplicity, we assume that each user  $A_i$  is able to perform one sensing task  $\kappa_i \in \mathcal{K}$ , which is common in many location-based crowdsensing systems due to the users' inability to appear in multiple places at the same time and limited sensing resources. However, this assumption can be readily relaxed. When  $A_i$  is able to perform multiple sensing tasks, we can convert  $A_i$  into multiple "virtual users" and each virtual user is responsible for exactly one sensing task that  $A_i$  can perform. Using this simple trick, all our methods and algorithms can be readily extended to the case that a user can perform multiple sensing tasks.

We assume that each user  $A_i$  has a *private value*  $\hat{d}_i \in \mathbb{R}^+$  indicating his/her sensing cost per unit time. For convenience, let  $\mu_i = u_{\kappa_i}$  denote the value of  $A_i$  to the owner for  $A_i$ 's one unit time sensing on task  $\kappa_i$ . We also assume that a user  $A_i$  is only available during the time period  $[\hat{s}_i, \hat{e}_i]$ , where  $\hat{s}_i, \hat{e}_i \in \mathbb{Z}$  are the earliest and latest available points in time private to  $A_i$ . Here  $\hat{s}_i$  and  $\hat{e}_i$  are both integers as they are defined with respect to certain time units. In reality, a user cannot be available all the time for sensing due to, for example, his/her own career. Therefore, we use an integer constant  $\lambda$  to denote the upper bound of  $(\hat{e}_i - \hat{s}_i), \forall i = 1, \dots, n$ ; the value of  $\lambda$  is called the (time) *interval range* of the users.

In a Mobile Crowdsensing Scheduling (MCS) problem (briefly illustrated in Fig. 1), the owner solicits the *bids*  $\vec{b} = (b_1, b_2, \dots, b_n)$  from the users in  $\mathcal{A}$ ; each  $b_i : 1 \leq i \leq n$  is a 3-tuple  $(d_i, s_i, e_i)$  where  $d_i \in \mathbb{R}^+$  and  $[s_i, e_i]$  ( $s_i, e_i \in \mathbb{Z}$ ) are  $A_i$ 's declared sensing cost (per unit time) and available time period, respectively. Let  $T_i = [s_i, e_i]$  for brevity. The owner then finds a *sensing time schedule*  $\vec{y}(\vec{b}) = (y_1(\vec{b}), y_2(\vec{b}), \dots, y_n(\vec{b}))$  for the users, where  $y_i(\vec{b}) \subseteq T_i$  is the time period allocated to  $A_i$  for sensing and it is not necessarily continuous. We also denote by  $|y_i(\vec{b})|$  the *total length* of  $y_i(\vec{b})$  in time. Based on the bids, the owner also

computes a *payment vector*  $\vec{p}(\vec{b}) = (p_1(\vec{b}), p_2(\vec{b}), \dots, p_n(\vec{b}))$ , where  $p_i(\vec{b}) \geq 0$  is the payment to  $A_i$  and  $\sum_{i=1}^n p_i(\vec{b}) \leq G$  should be satisfied. Moreover, the payment to any user  $A_i$  should be no less than his/her total sensing cost if all users bid truthfully, i.e.,  $p_i(\vec{b}) \geq d_i |y_i(\vec{b})|$ . Defining the owner's *revenue*,  $R(\vec{y}(\vec{b}))$ , as the total sensing value of performing the sensing tasks allocated by the sensing schedule  $\vec{y}(\vec{b})$ , i.e.,

$$R(\vec{y}(\vec{b})) = \sum_{i=1}^m u_i \cdot \left| \left| \bigcup_{j:\kappa_j=i} y_j(\vec{b}) \right| \right|,$$

the goal of the MCS problem is to maximize this revenue subject to all the above constraints. Note that should we assume that the users always bid truthfully, the MCS problem would become a pure combinatorial optimization problem. As proved by *Theorem 1*, this simplified problem is NP-hard.

*Theorem 1:* The MCS problem with all users bidding truthfully is NP-hard.

For notational simplicity, we sometimes omit  $\vec{b}$  when writing the schedule and payment vectors (e.g., writing  $\vec{y}$  instead of  $\vec{y}(\vec{b})$ ), if the bid  $\vec{b}$  is clear from the context.

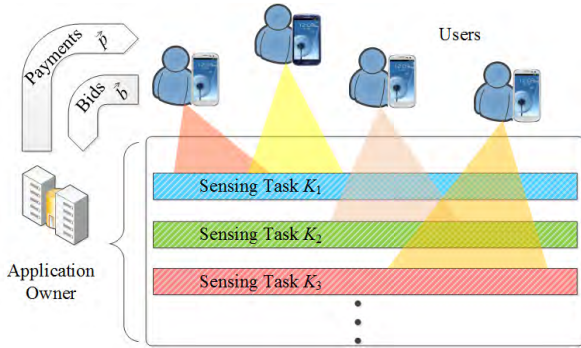


Fig. 1. The mobile crowdsensing scheduling (MCS) problem.

## 2.2 Offline and Online Truthful Mechanisms

Each user  $A_i$  has a *utility* indicating the difference between the payment made to him/her and his/her total sensing cost according to the sensing schedule, i.e.,  $p_i(\vec{b}) - \hat{d}_i |y_i(\vec{b})|$ . In practice, the users are selfish and are only interested in maximizing their own utilities. For this purpose, they may bid strategically, i.e., lying about their private information such as sensing costs and available time. To handle users' strategic behaviors, we need to design *truthful* mechanisms for the MCS problem to align the users' interests with the system goal of revenue maximization. A mechanism is called (dominant-strategy) truthful if any user maximizes his/her utility by revealing his/her real private information, no matter how other users may act [8]. A randomized mechanism is called truthful (or *universally-truthful*) if it is a randomization over a set of truthful mechanisms. Moreover, we also require our mechanisms to satisfy *individual*

*rationality* (IR here after), which means that any truth-telling user  $A_i$  always gets a non-negative utility [8], i.e.,  $p_i((\hat{d}_i, \hat{s}_i, \hat{e}_i), b_{-i}) \geq \hat{d}_i |y_i((\hat{d}_i, \hat{s}_i, \hat{e}_i), b_{-i})|$ , where  $b_{-i}$  represents the bids of the users other than  $A_i$ .

In the following sections, we aim to design truthful mechanisms for the MCS problem under both the offline and online settings. In the offline setting, the owner collects all the users' bids before scheduling them, which corresponds to a practical scenario that the users reserve sensing tasks in advance. In the online setting, the users' bids are revealed one by one, and the owner must make an irrevocable decision on scheduling any user right at the moment when the user's bid is revealed. This setting corresponds to another practical scenario where the users arrive randomly at some sensing area, and we assume that the users' arrival order is drawn uniformly at random from the set of all permutations over the users. We will design five algorithms for the offline and online settings, and the call graph of our algorithms is shown in Fig. 2, which illustrates the overall structure of our mechanisms proposed in Sec. 3 - Sec. 5.

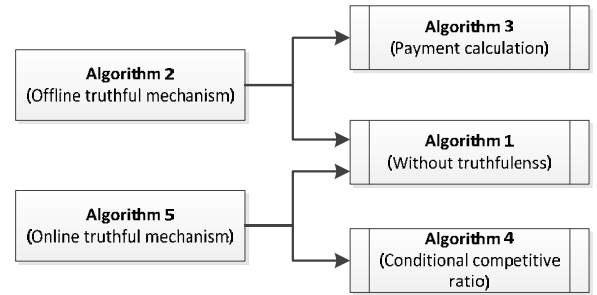


Fig. 2. The call graph of our algorithms.

## 3 APPROXIMATION ALGORITHMS FOR MCS

In this section, we treat the MCS problem as a pure combinatorial optimization problem and design approximation algorithms for it, as shown in *Algorithm 1*. Although the strategic behaviors of the users are not considered in *Algorithm 1*, this algorithm serves as an important building block for the truthful mechanisms designed later.

A partial order  $\prec$  on the set  $\mathcal{A}$  is used in *Algorithm 1*, which is defined as follows. For any two users  $A_i$  and  $A_j$ , if  $\mu_i/d_i < \mu_j/d_j$  or  $\mu_i/d_i = \mu_j/d_j$  but  $j > i$ , then we say  $A_j$  *suppress*  $A_i$  and denote it by  $A_i \prec A_j$ . For any  $\mathcal{A}' \subseteq \mathcal{A}$ , we define  $\max_{\prec}(\mathcal{A}')$  to be the user in  $\mathcal{A}'$  such that there does not exist another user  $\hat{A} \in \mathcal{A}'$  satisfying  $\max_{\prec}(\mathcal{A}') \prec \hat{A}$ . *Algorithm 1* iterates among the users and finds the schedule for them based on a greedy strategy. The algorithm, at the beginning of each iteration, selects a user  $A_j$  based on the partial order  $\prec$  (line 5), and then computes the time units that can potentially be scheduled for  $A_j$  (line 6). The sensing time  $y_j(\vec{b})$  scheduled for  $A_j$  is taken as the early sub-period of the *uncovered* time of  $A_j$ , where any time point is called

**Algorithm 1:** ApproxMCS( $n, G, \vec{b}, \mathcal{A}$ )

---

```

1  $\mathcal{D} \leftarrow \mathcal{A}; \mathcal{W} \leftarrow \emptyset;$ 
2 for  $i \leftarrow 1$  to  $n$  do  $y_i(\vec{b}) \leftarrow \emptyset$ 
3 ;
4 repeat
5   Find  $j$  such that  $A_j = \max_{\prec}(\mathcal{D})$ 
6    $Z_j \leftarrow T_j - \bigcup_{i:\kappa_i=\kappa_j} y_i(\vec{b})$ 
7    $q \leftarrow \min \left\{ |Z_j|, \left\lfloor \frac{G}{2d_j} - \frac{R(\vec{y}(\vec{b}))}{\mu_j} \right\rfloor \right\}$ 
8   if  $q > 0$  then
9      $\mathcal{W} \leftarrow \mathcal{W} \cup \{j\}$ 
10     $y_j(\vec{b}) \leftarrow$  The first  $q$  time units in  $Z_j$ 
11     $\mathcal{D} \leftarrow \mathcal{D} - \{A_i \in \mathcal{A} | T_i \subseteq \bigcup_{\ell:\kappa_\ell=\kappa_i} y_\ell(\vec{b})\}$ 
12 until  $q < |Z_j|$  or  $\mathcal{D} = \emptyset;$ 
13 for  $i \leftarrow 1$  to  $n$  do  $p_i(\vec{b}) \leftarrow d_i \cdot |y_i(\vec{b})|$ 
14 ;
15 return  $\vec{y}(\vec{b}), \mathcal{W}, \vec{p}(\vec{b})$ 

```

---

uncovered if no user has been scheduled for it (line 10). At the end of an iteration, all users whose available time periods have been covered are removed from the user set  $\mathcal{D}$  upon which the algorithm iterates (line 11). The algorithm determines the length of  $y_j(\vec{b})$  based on the rule in line 7, which can be deemed as a *potential function* [12] that facilitates our later quest for an approximation ratio. Also, the rule serves as a constraint to bound the total payments below  $G$ . Finally, the payment to each user  $A_i$  is simply calculated as the product of  $d_i$  and the length of  $y_i(\vec{b})$  (line 13), as the strategic behaviours of the users are not considered in **Algorithm 1**.

Let us denote an iteration (from line 4 to 12) in which line 10 is executed as an *effective iteration* (i.e., the concerned user is assigned a non-empty schedule). Suppose that **Algorithm 1** has in total  $h$  effective iterations. Let the user scheduled in the  $i$ th effective iteration be  $A_{n_i} : 1 \leq i \leq h$  and let  $\mathcal{W} = \{n_1, n_2, \dots, n_h\}$  be the *index set* of these scheduled users. Let  $\vec{y}_i(\vec{b})$  be the current value of vector  $\vec{y}(\vec{b})$  after the  $i$ th effective iteration is executed, we have the following results.

**Theorem 2:** The output of **Algorithm 1** is a feasible solution to the MCS problem.

**Theorem 3:** Let  $\vec{y}^*$  be an optimal schedule vector for the MCS problem and  $\Lambda = \max_{i:d_i \leq G} (\mu_i \cdot |T_i|)$ . If

$$\Lambda \leq \left[ \frac{e-1}{4e} - \epsilon \right] R(\vec{y}^*), \quad (1)$$

for any  $\epsilon \in (0, \frac{e-1}{4e})$ , then  $\vec{y}$  is a  $\frac{4}{3}\epsilon$  approximation to MCS.

Note that an extra condition (1) needs to be satisfied in **Theorem 3**, hence the approximation ratio is conditional. We shall handle this issue using a randomized mechanism design method in the next section.

## 4 OFFLINE MECHANISMS FOR MCS

As users' strategic behaviors are not considered in **Algorithm 1**, the payments to all users made there are directly determined by the declared sensing cost and the length of the time periods scheduled for them. Unfortunately, this method can be non-truthful: a user may lie about his/her private information to manipulate the length of his/her scheduled time, hence to gain a higher utility. Characterizations of truthful mechanisms exist in the literature (e.g., [13], [14]), but these characterizations are only for single parameter mechanisms, while in our problem any user  $A_i$  has three parameters, namely  $d_i$ ,  $s_i$  and  $e_i$ . Therefore, we hereby design a novel truthful mechanism for the MCS problem under the offline setting, as shown by **Algorithm 2**. A sub-routine used by **Algorithm 2** to compute payments to users is shown in **Algorithm 3**.

**Algorithm 2:** Truthful Offline Mechanism for MCS

---

```

1 Generate a random number  $o$  from the uniform
  distribution on the interval  $[0, 1]$ 
2 if  $o \leq \frac{1}{2}$  then
3    $(\vec{y}(\vec{b}), \mathcal{W}) \leftarrow$  ApproxMCS( $n, G, \vec{b}, \mathcal{A}$ )
4   forall the  $n_i \in \mathcal{W}$  do
5      $p_{n_i}(\vec{b}) \leftarrow$  CalPayment( $n, G, \vec{b}, \mathcal{A}, \mathcal{W}, \vec{y}, n_i$ )
6   forall the  $i \notin \mathcal{W}$  do  $p_i(\vec{b}) \leftarrow 0;$ 
7 else
8   forall the  $i \in \mathcal{A}$  do  $p_i(\vec{b}) \leftarrow 0$ 
9   ;
10   $j \leftarrow \arg \max_{i:d_i \leq G} \mu_i; \mathcal{W} \leftarrow \{j\}$ 
11   $y_j(\vec{b}) \leftarrow$  an arbitrary time unit in  $T_j; p_j(\vec{b}) \leftarrow G$ 
12 Pay  $p_i(\vec{b})$  to  $A_i : \forall i \in \mathcal{W}$  at the end of  $T_i$  if  $A_i$ 
  successfully completes his/her sensing task during
   $y_i(\vec{b})$ 
13 return  $\vec{y}(\vec{b}), \vec{p}(\vec{b})$ 

```

---

**Algorithm 2** is apparently a randomized mechanism. With probability one half, the algorithm calls **Algorithm 1** to get a feasible schedule for each user (line 3). However, instead of using the simple payment rule in **Algorithm 1**, **Algorithm 2** replaces it with a more complicated method shown by **Algorithm 3** to calculate the payments (line 5); otherwise **Algorithm 2** runs lines 8-11 and selects a user whose sensing cost per unit time is no more than the budget and whose sensing value per unit time is maximized (see line 10). Then the selected user is paid the amount  $G$ , while others are paid zero. The payments are made to the users using a *post-paid* scheme, i.e., a payment is made instantly at the end of a user's claimed available time period only if he/she has successfully performed the sensing task during the whole time period scheduled for him/her (line 12). To understand the payment calculation in **Algorithm 3**, we introduce **Lemma 1**, **Lemma 2** and **Theorem 4**, which

**Algorithm 3:** CalPayment( $n, G, \vec{b}, \mathcal{A}, \mathcal{W}, \vec{y}, n_i$ )

---

```

1 for  $j \leftarrow 1$  to  $n$  do  $t_j \leftarrow \emptyset$ 
2 ;
3 for  $j \leftarrow 1$  to  $i-1$  do  $t_{n_j} \leftarrow y_{n_j}(\vec{b})$ 
4 ;
5  $\mathcal{S} \leftarrow \{j \mid 1 \leq j \leq n, \frac{\mu_j}{d_j} < \frac{\mu_{n_i}}{d_{n_i}}\}$ 
6  $p_{n_i}(\vec{b}) \leftarrow d_{n_i} \cdot |y_{n_i}(\vec{b})|$ ;  $k \leftarrow n_i$ ;  $\theta \leftarrow |\mathcal{S}|$ 
7 while  $\theta \geq 0$  do
8    $\mathcal{Z}_1 \leftarrow T_{n_i} - \bigcup_{\ell: \kappa_{n_i} = \kappa_\ell} t_\ell$ ;  $\gamma_1 \leftarrow \frac{\mu_{n_i}}{\mu_k} \cdot d_k$ ;  $\gamma_2 \leftarrow \frac{\mu_{n_i} G}{2R(\vec{t})}$ 
9   if  $\mathcal{S} \neq \emptyset$  then
10    Find  $j$  such that  $A_j = \max_{\prec}(\mathcal{S})$ 
11     $\gamma_2 \leftarrow \min \left\{ \frac{\mu_{n_i}}{\mu_j} \cdot d_j, \gamma_2 \right\}$ 
12    if  $|\mathcal{Z}_1| > 0 \wedge \gamma_2 \geq \gamma_1$  then
13      $p_{n_i}(\vec{b}) = p_{n_i}(\vec{b}) + \int_{\gamma_1}^{\gamma_2} \min \left\{ |\mathcal{Z}_1|, \left[ \frac{G}{2v} - \frac{R(\vec{t})}{\mu_{n_i}} \right] \right\} dv$ 
14    else break;
15
16    if  $\mathcal{S} \neq \emptyset$  then
17      $\mathcal{Z}_2 \leftarrow T_j - \bigcup_{\ell: \kappa_j = \kappa_\ell} t_\ell$ 
18      $q \leftarrow \min \left\{ |\mathcal{Z}_2|, \left[ \frac{G}{2d_j} - \frac{R(\vec{t})}{\mu_j} \right] \right\}$ 
19     if  $q > 0$  then
20       $t_j \leftarrow$  The first  $q$  time units in  $\mathcal{Z}_2$ 
21     if  $q < |\mathcal{Z}_2|$  then break;
22
23      $k \leftarrow j$ ;  $\theta \leftarrow |\mathcal{S}|$ ;  $\mathcal{S} \leftarrow \mathcal{S} \setminus \{j\}$ 
24    $\theta \leftarrow \theta - 1$ 
25 return  $p_{n_i}(\vec{b})$ 

```

---

are also useful for characterizing truthfulness under our multi-parameter environment. The detailed explanation of **Algorithm 3** can be found in the proof of **Theorem 4**.

**Lemma 1:** For any user  $A_i$  and his/her two bids  $b_i = (d_i, s_i, e_i)$  and  $b'_i = (d'_i, s_i, e_i)$ , if  $d'_i \geq d_i$ , then  $|y_i(b'_i, b_{-i})| \leq |y_i(b_i, b_{-i})|$ .

**Lemma 2:** For any user  $A_i$  and his/her two bids  $b_i = (d_i, s_i, e_i)$  and  $b'_i = (d_i, s'_i, e'_i)$ , if  $[s'_i, e'_i] \subseteq [s_i, e_i]$ , then  $|y_i(b'_i, b_{-i})| \leq |y_i(b_i, b_{-i})|$ .

**Theorem 4:** For any user  $A_i$  with the bid  $b_i = (d_i, s_i, e_i)$  and any  $b_{-i}$ ,  $\int_0^\infty |y_i((v, s_i, e_i), b_{-i})| dv < +\infty$  and the payment to  $A_i$  computed by **Algorithm 3** is

$$p_i(b_i, b_{-i}) = d_i |y_i(b_i, b_{-i})| + \int_{d_i}^\infty |y_i((v, s_i, e_i), b_{-i})| dv. \quad (2)$$

*Proof:* For  $i \notin \{n_1, n_2, \dots, n_h\}$ , we have  $p_i(b_i, b_{-i}) = 0$  and  $|y_i(b_i, b_{-i})| = 0$ . According to **Lemma 1**, for any  $v > d_i$ , we have  $|y_i((v, s_i, e_i), b_{-i})| \leq |y_i(b_i, b_{-i})|$ , hence  $|y_i((v, s_i, e_i), b_{-i})| = 0$ . So (2) holds in this case.

Now we analyze **Algorithm 3** in details for the case of  $i \in \{n_1, n_2, \dots, n_h\}$ , and we write  $i$  as  $n_i$  in this case. In **Algorithm 3**, we first initialize the vector  $\vec{t}$  to record the user schedules that are decided before  $A_{n_i}$ , as shown in lines 1-3. Then we use  $\mathcal{S}$  to denote the indices of

the users that are suppressed by  $A_{n_i}$  according to the partial order  $\prec$  (line 5). Note that if  $A_{n_i}$  bids  $d'_{n_i} > d_{n_i}$ , the uncovered available time period of  $A_{n_i}$  (i.e.,  $\mathcal{Z}_1$ ) may change, because we have  $\mu_{n_i}/d'_{n_i} < \mu_{n_i}/d_{n_i}$  and some users originally suppressed by  $A_{n_i}$  may hence get scheduled before  $A_{n_i}$ . Consequently, we calculate the schedule for other users (recorded in  $\vec{t}$ ) when  $d_{n_i}$  increases, and divide the interval  $[d_{n_i}, +\infty)$  into some sub-intervals, such that the uncovered available time period of  $A_{n_i}$  remains the same when  $A_{n_i}$  changes  $d_{n_i}$  within each of these sub-intervals, as shown in lines 7-24. More specifically, the algorithm, at the beginning of each iteration, picks a user's index  $j$  from  $\mathcal{S}$  such that  $A_j$  is maximal with respect to the partial order  $\prec$  (line 10), then it identifies a sub-interval  $\left( \frac{\mu_{n_i}}{\mu_k} \cdot d_k, \frac{\mu_{n_i}}{\mu_j} \cdot d_j \right]$  for  $d_{n_i}$  (indicated by  $\gamma_1$  and  $\gamma_2$ , see line 8 and line 11), where  $k$  is the index of the last picked user from  $\mathcal{S}$  (line 23, but initially set as  $n_i$ ). As  $A_{n_i}$  remains scheduled before  $A_j$  when  $d_{n_i}$  varies within this sub-interval, we calculate the partial payment in line 13 based on the current schedule. When  $d_{n_i}$  gets bigger than  $\frac{\mu_{n_i}}{\mu_j} \cdot d_j$ , then  $A_j$  will be scheduled before  $A_{n_i}$ , and we calculate  $A_j$ 's schedule in this case by lines 16-23, where the length of time slots allocated to  $A_j$  is not allowed to exceed the length of  $A_j$ 's unallocated time slots (line 18). With this adjusted schedule, the algorithm goes into the next iteration to further accumulate the partial payment in a different sub-interval. At the end, the algorithm goes through all the possible sub-intervals in  $[d_{n_i}, \infty)$ , so the payment is exactly calculated as the right-hand side of (2). Hence the theorem follows.  $\square$

In **Lemma 3** and **Theorem 5**, we prove that the payment calculated by **Algorithm 3** is no more than the budget  $G$  and **Algorithm 2** provides a feasible solution satisfying IR to the MCS problem.

**Lemma 3:** For any user  $A_j$  and his/her bid  $(d_j, s_j, e_j)$ , if  $j \in \{n_1, n_2, \dots, n_h\}$ , then we have  $d_j \leq \mu_j \cdot G/R(\vec{y})$ .

**Theorem 5:** The mechanism shown in **Algorithm 2** provides a feasible solution that satisfies IR to the MCS problem.

More importantly, using the results stated in **Lemma 1**, **Lemma 2** and **Theorem 4**, we can now prove the truthfulness of the mechanism by **Theorem 6**. The rationale lies in the difference between any user  $A_i$ 's payment and his/her true sensing cost: regardless of how other users may bid, this difference is always maximized if  $A_i$  bids truthfully and hence truth-telling is a dominant strategy for  $A_i$ .

**Theorem 6:** The scheduling mechanism shown in **Algorithm 2** is truthful.

*Proof:* We first prove that lines 3-6 is truthful. Suppose that there exists a user  $A_i$  whose truthful bid is  $\hat{b}_i = (\hat{d}_i, \hat{s}_i, \hat{e}_i)$ , but he/she can get a higher utility by bidding  $b_i = (d_i, s_i, e_i) \neq \hat{b}_i$  for some  $b_{-i}$ . If  $y_i(b_i, b_{-i}) \not\subseteq [\hat{s}_i, \hat{e}_i]$ ,  $A_i$  gets zero payment because the mechanism requires  $A_i$  to complete sensing during  $y_i(b_i, b_{-i})$  to get paid. If  $e_i \notin [\hat{s}_i, \hat{e}_i]$ ,  $A_i$  again gets zero payment because the

mechanism employs a post-paid scheme and  $A_i$  cannot get paid when he/she is unavailable. Therefore, we must have  $y_i(b_i, b_{-i}) \subseteq [\hat{s}_i, \hat{e}_i]$  and  $e_i \in [\hat{s}_i, \hat{e}_i]$ . Now if  $s_i \geq \hat{s}_i$ , we have  $[s_i, e_i] \subseteq [\hat{s}_i, \hat{e}_i]$ ; so using **Lemma 2** we get

$$|y_i((d_i, s_i, e_i), b_{-i})| \leq |y_i((d_i, \hat{s}_i, \hat{e}_i), b_{-i})|. \quad (3)$$

Otherwise if  $s_i < \hat{s}_i$ , we know that the period  $[s_i, \hat{s}_i]$  must have been covered before deciding the schedule of  $A_i$  based on his/her bidding  $b_i$ , because otherwise the algorithm will allocate time in  $[s_i, \hat{s}_i]$  to  $A_i$  according to line 10 of **Algorithm 1**, which contradicts  $y_i(b_i, b_{-i}) \subseteq [\hat{s}_i, \hat{e}_i]$ . Hence we know  $y_i((d_i, s_i, e_i), b_{-i}) = y_i((d_i, \hat{s}_i, e_i), b_{-i})$ . As  $[s_i, e_i] \subseteq [\hat{s}_i, \hat{e}_i]$ , (3) also holds by using **Lemma 2**.

For  $A_i$ 's any bid  $(d'_i, s'_i, e'_i)$ , let  $f_i((d'_i, s'_i, e'_i), b_{-i})$  denote the uncovered time in  $[s'_i, e'_i]$  when the algorithm allocates time to  $A_i$  based on a bid vector  $((d'_i, s'_i, e'_i), b_{-i})$ . The above reasoning actually reveals that  $f_i((d_i, s_i, e_i), b_{-i}) \subseteq f_i((d_i, \hat{s}_i, \hat{e}_i), b_{-i})$ . According to the mechanism, for any  $v \geq d_i$  we have  $f_i((v, s_i, e_i), b_{-i}) \subseteq f_i((d_i, s_i, e_i), b_{-i})$ ,  $f_i((v, \hat{s}_i, \hat{e}_i), b_{-i}) \subseteq f_i((d_i, \hat{s}_i, \hat{e}_i), b_{-i})$  and

$$\begin{aligned} [f_i((d_i, \hat{s}_i, \hat{e}_i), b_{-i}) \setminus f_i((v, \hat{s}_i, \hat{e}_i), b_{-i})] \cap f_i((d_i, s_i, e_i), b_{-i}) \\ \subseteq f_i((d_i, s_i, e_i), b_{-i}) \setminus f_i((v, s_i, e_i), b_{-i}), \end{aligned}$$

which yield  $f_i((v, s_i, e_i), b_{-i}) \subseteq f_i((v, \hat{s}_i, \hat{e}_i), b_{-i})$  and

$$|y_i((v, s_i, e_i), b_{-i})| \leq |y_i((v, \hat{s}_i, \hat{e}_i), b_{-i})|. \quad (4)$$

Since the user gets more utility by bidding  $b_i$  then by bidding  $\hat{b}_i$ , we know that:

$$\begin{aligned} p_i((d_i, s_i, e_i), b_{-i}) - \hat{d}_i \cdot |y_i((d_i, s_i, e_i), b_{-i})| \\ > p_i((\hat{d}_i, \hat{s}_i, \hat{e}_i), b_{-i}) - \hat{d}_i \cdot |y_i((\hat{d}_i, \hat{s}_i, \hat{e}_i), b_{-i})|. \end{aligned}$$

Combing this with **Theorem 4** gives us

$$\begin{aligned} (d_i - \hat{d}_i) \cdot |y_i((d_i, s_i, e_i), b_{-i})| \\ > \int_{\hat{d}_i}^{\infty} |y_i((v, \hat{s}_i, \hat{e}_i), b_{-i})| dv - \int_{d_i}^{\infty} |y_i((v, s_i, e_i), b_{-i})| dv. \end{aligned}$$

Case 1:  $d_i \geq \hat{d}_i$ , using (4) and **Lemma 1** we get

$$\begin{aligned} (d_i - \hat{d}_i) \cdot |y_i((d_i, s_i, e_i), b_{-i})| \\ > \int_{\hat{d}_i}^{d_i} |y_i((v, \hat{s}_i, \hat{e}_i), b_{-i})| dv \\ \geq (d_i - \hat{d}_i) \cdot |y_i((d_i, \hat{s}_i, \hat{e}_i), b_{-i})|. \end{aligned}$$

If  $d_i = \hat{d}_i$ , then we get  $0 > 0$ , a contradiction. If  $d_i > \hat{d}_i$ , then we get  $|y_i((d_i, s_i, e_i), b_{-i})| > |y_i((d_i, \hat{s}_i, \hat{e}_i), b_{-i})|$ , which contradicts (3).

Case 2:  $d_i < \hat{d}_i$ , using (4) and **Lemma 1** we get:

$$\begin{aligned} (d_i - \hat{d}_i) \cdot |y_i((d_i, s_i, e_i), b_{-i})| \\ > - \int_{d_i}^{\hat{d}_i} |y_i((v, s_i, e_i), b_{-i})| dv \\ \geq (d_i - \hat{d}_i) \cdot |y_i((d_i, s_i, e_i), b_{-i})|, \end{aligned}$$

hence  $|y_i((d_i, s_i, e_i), b_{-i})| < |y_i((d_i, s_i, e_i), b_{-i})|$ , also a contradiction.

The above reasoning has shown that lines 3-6 is truthful. Now we prove that lines 8-11 is truthful. If a user  $A_i$  gets a non-empty schedule by bidding truthfully, then  $A_i$  clearly cannot benefit from lying. Now suppose that  $A_i$  gets an empty schedule (hence the utility 0) by bidding truthfully. If  $\hat{d}_i \leq G$ , then  $A_i$  cannot increase his/her utility by lying, because he/she will anyway get an empty schedule regardless of his/her bid. If  $\hat{d}_i > G$ , then the only way that may allow  $A_i$  to get a non-empty schedule is to bid some  $(d_i, s_i, e_i)$  with  $d_i \leq G$ . However, in that case  $A_i$ 's utility is  $G - \hat{d}_i < 0$ . Therefore,  $A_i$  is better off bidding truthfully. From the above reasoning, we know that **Algorithm 2** is a randomization of two truthful mechanisms, and is hence truthful.  $\square$

Finally, based on the approximation ratio of **Algorithm 1**, we can prove that **Algorithm 2** has an  $\mathcal{O}(1)$  approximation ratio, as shown by **Theorem 7**. We also analyze the time complexity of **Algorithm 2** by **Theorem 8**.

**Theorem 7:** The mechanism shown in **Algorithm 2** has an approximation ratio of  $\mathcal{O}(1)$ .

**Theorem 8:** The worst-case time complexity of **Algorithm 2** is  $\mathcal{O}(n^2)$ .

Obviously, the major time complexity results from calling **Algorithm 3** for at most  $n$  times.

## 5 ONLINE MECHANISMS FOR MCS

In this section, we study incentive mechanisms for the MCS problem under the online setting, where the users come in random orders and the schedule/payment for each user has to be decided upon his/her arrival. We assume in this case that any user  $A_i$  would only lie about his/her sensing cost  $\hat{d}_i$ , and we will design truthful mechanisms such that reporting his/her real cost is a dominant strategy of  $A_i$ . The problem of handling users' strategic bidding on their available time periods under the online setting is left for future work.

We first propose a simple online mechanism in **Algorithm 4** for the MCS problem, whose idea originates from the *secretary algorithm* [15]. In lines 5-6 of **Algorithm 4**, we assign empty schedules to the first arrived  $\lfloor \frac{n}{e} \rfloor$  users, and find one of them whose sensing cost per unit time is no more than the budget and whose sensing value per unit time is the maximum denoted by  $\alpha$  (see line 6). The value of  $\alpha$  is then used as a threshold for the later users, among which we will select the first one whose sensing value per unit time is no less than  $\alpha$  and pay him/her  $G$  (lines 8-10); other users all get empty schedules and zero payments (line 12). The schedules and payments assigned to the users are returned by vector  $\vec{y}$  and vector  $\vec{p}$  (line 13), respectively.

Clearly, **Algorithm 4** provides a feasible solution to the MCS problem and satisfies IR. The truthfulness and competitive ratio of **Algorithm 4** are given in **Theorem 9** and **Theorem 10**, respectively:

**Theorem 9:** The online scheduling mechanism in **Algorithm 4** is truthful.

**Algorithm 4: A Deterministic Online Mechanism**


---

```

1  $\alpha \leftarrow 0; j \leftarrow 0$ 
2 upon  $A_i$ 's arrival
3    $j \leftarrow j + 1$ 
4   if  $j \leq \lfloor \frac{n}{e} \rfloor$  then
5      $y_i \leftarrow \emptyset; p_i \leftarrow 0$ 
6     if  $d_i \leq G$  then  $\alpha \leftarrow \max\{\alpha, \mu_i\};$ 
7   else
8     if  $\mu_i \geq \alpha \wedge d_i \leq G \wedge G > 0$  then
9        $y_i \leftarrow$  an arbitrary time unit in  $T_i$ 
10       $p_i \leftarrow G; G \leftarrow 0$ 
11     else
12        $y_i \leftarrow \emptyset; p_i \leftarrow 0$ 
13 return  $(\vec{y}, \vec{p})$ 

```

---

*Proof:* As the users cannot control their arrival sequence, they also cannot control the value of  $\alpha$ . The first arrived  $\lfloor \frac{n}{e} \rfloor$  users are always assigned the empty schedule, so they always get the utility 0 no matter how they bid. Now consider any  $A_i$  whose arrival order is greater than  $\lfloor \frac{n}{e} \rfloor$ . If  $A_i$  gets a non-empty schedule by bidding  $\hat{d}_i$ , then it is clear that he/she cannot benefit from lying, because his/her utility  $G - \hat{d}_i \geq 0$  is the largest one he/she can possibly get. Otherwise if  $A_i$  gets an empty schedule by bidding  $\hat{d}_i$ , then there are two cases we need to consider: (i)  $G = 0$  when  $A_i$  arrives: In this case,  $A_i$  will always get utility 0 no matter how he/she bids. (ii)  $G > 0$  when  $A_i$  arrives: In this case we must have  $\mu_i < \alpha$  or  $\hat{d}_i > G$ . If  $\mu_i < \alpha$ , then  $A_i$  always gets utility 0 regardless of his/her bid. If  $\mu_i \geq \alpha$  and  $\hat{d}_i > G$ , bidding any  $d_i > G$  will always cause  $A_i$  to get an empty schedule (hence the utility 0), while bidding  $d_i \leq G$  would enable  $A_i$  to get a non-empty schedule, but his/her utility  $G - \hat{d}_i$  would be negative; hence he/she is better off bidding truthfully.  $\square$

**Theorem 10:** If  $\Lambda \geq \frac{R(\vec{y}^*)}{150}$  (where  $\Lambda$  was defined in **Theorem 3**), then **Algorithm 4** has a  $\mathcal{O}(1)$  competitive ratio with a constant probability.

Note that the competitive ratio stated in **Theorem 10** is conditional. To rectify this problem, we propose a randomized mechanism shown in **Algorithm 5**, which runs **Algorithm 4** with probability one half and runs lines 3-16 otherwise. Roughly speaking, the idea of lines 3-16 is the following: we assign the empty schedule to the first arrived  $\xi$  users and use them as a random sample to guess the optimal solution  $R(\vec{y}^*)$  in lines 7-10, where **Algorithm 1** is called to calculate the sensing schedule vector  $\vec{r}$  for the  $\xi$  users that have arrived (line 10); then we use this guess (i.e.,  $R(\vec{r})$ ) to schedule the users coming afterwards (lines 12-16). More specifically, line 12 calculates a threshold  $\eta$  based on the total budget  $G$  and the estimated optimal value  $R(\vec{r})$ , and lines 13-16 use this threshold to schedule the users according to the current budget  $M$ . It can be seen from lines 13-14 that

**Algorithm 5: A Randomized Online Mechanism**


---

```

1 Generate a random number  $o$  from the uniform
  distribution on the interval  $[0, 1]$ ;
2 if  $o \leq \frac{1}{2}$  then
3   Let  $\xi$  be a random number generated from the
  binomial distribution  $B(n, 1/2)$ 
4    $M \leftarrow G; j \leftarrow 0; \mathcal{A}_\xi \leftarrow \emptyset$ 
5   upon  $A_i$ 's arrival
6      $j \leftarrow j + 1$ 
7     if  $j \leq \xi$  then
8        $y_i \leftarrow \emptyset; p_i \leftarrow 0; \mathcal{A}_\xi \leftarrow \mathcal{A}_\xi \cup \{A_i\}$ 
9       if  $j = \xi$  then
10         $\vec{r}(\vec{b}_\xi) \leftarrow \text{ApproxMCS}(\xi, G, \vec{b}_\xi, \mathcal{A}_\xi)$ 
11      else
12         $\eta \leftarrow 5G \cdot \mu_i / R(\vec{r}); F_i \leftarrow T_i - \bigcup_{\ell: \kappa_i = \kappa_\ell} y_\ell$ 
13        if  $d_i \leq \eta \wedge \eta \cdot |F_i| \leq M \wedge |F_i| \geq 1$  then
14           $y_i \leftarrow F_i; p_i \leftarrow \eta \cdot |F_i|; M \leftarrow M - p_i$ 
15        else
16           $y_i \leftarrow \emptyset; p_i \leftarrow 0$ 
17 else Run Algorithm 4 to get  $(\vec{y}, \vec{p})$ 
18 ;
19 return  $(\vec{y}, \vec{p})$ 

```

---

**Algorithm 5** satisfies IR and provides a feasible solution to the MCS problem. The truthfulness of **Algorithm 5** is proven in **Theorem 11**:

**Theorem 11:** The online scheduling mechanism shown in **Algorithm 5** is truthful.

*Proof:* We have proved in **Theorem 9** that **Algorithm 4** is truthful, so we only need to prove that lines 3-16 are truthful given that users strategically report their sensing costs. Note that the users cannot control their arrival order as well as the value of  $R(\vec{r})$ , hence the first arrived  $\xi$  users always get utility 0 no matter how they bid. Now consider any user  $A_i$  who arrives afterwards. If  $A_i$  gets a non-empty schedule (i.e.,  $|y_i| \neq 0$ ) by bidding  $\hat{d}_i$ , then we know that  $\hat{d}_i \leq \eta$  and  $A_i$  gets the utility  $(\eta - \hat{d}_i) \cdot |F_i| \geq 0$ , which remains the same if  $A_i$  bids any  $d_i \leq \eta$ . If  $A_i$  bids  $d_i > \eta$ , then his/her utility will be 0, so he/she is better off bidding his/her true value. Otherwise if  $|y_i| = 0$  when  $A_i$  bids  $\hat{d}_i$ , then at least one of the following conditions holds: (i)  $\eta \cdot |F_i| > M$ ; (ii)  $|F_i| = 0$ ; (iii)  $\hat{d}_i > \eta$ . If (i) or (ii) holds,  $A_i$ 's utility remains 0 no matter how he/she bids. If (iii) holds, then bidding  $d_i \leq \eta$  may get  $A_i$  assigned a non-empty schedule, but in that case the utility of  $A_i$  would be  $(\eta - \hat{d}_i) \cdot |F_i| < 0$ , hence he/she is still better off bidding the true value  $\hat{d}_i$ .  $\square$

Finally, the competitive ratio and time complexity of **Algorithm 5** are given in **Lemma 4-6** and **Theorem 12**:

**Lemma 4:** Let  $(\sigma_1, \sigma_2, \dots, \sigma_n)$  be the actual arrival sequence of the users' indices, which is a permutation of  $\{1, 2, \dots, n\}$ . Let  $\Delta_1 = \sum_{i=1}^{\xi} \mu_{\sigma_i} \cdot |y_{\sigma_i}^*|$  and  $\Delta_2 =$

$\sum_{i=\xi+1}^n \mu_{\sigma_i} \cdot |y_{\sigma_i}^*|$ . If  $\Lambda \leq \frac{R(\bar{y}^*)}{150}$ , then  $\Delta_1 \geq R(\bar{y}^*)/3$  and  $\Delta_2 \geq R(\bar{y}^*)/4$  hold at the same time with constant probability.

**Lemma 5:** If  $\Lambda \leq \frac{R(\bar{y}^*)}{150}$ , then  $\frac{\Delta_1}{5} \leq R(\bar{r}) \leq R(\bar{y}^*)$ .

**Lemma 6:** If  $\Lambda \leq \frac{R(\bar{y}^*)}{150}$ , then the solution output by lines 3-16 of **Algorithm 5** has an  $\mathcal{O}(1)$  competitive ratio with constant probability.

**Theorem 12:** The competitive ratio and worst-case time complexity of **Algorithm 5** are  $\mathcal{O}(1)$  and  $\mathcal{O}(n^2)$ , respectively.

## 6 SIMULATIONS

We conduct extensive simulations to evaluate the performance of our truthful scheduling mechanisms. The objective of our simulations is to corroborate the correctness and effectiveness of our mechanisms in terms of various metrics (including owner revenue, total payment, truthfulness and IR) under different parameter settings (such as the number of users or tasks, as well as the budget). Since we are the first, to the best of our knowledge, to study the MCS problem (see the discussions in Sec. 1 and the definition in Sec. 2), we can only make comparisons between our own algorithms in the simulations. For brevity, we hereby denote our two main algorithms **Algorithm 2** and **Algorithm 5** by “offline” and “online”, respectively.

### 6.1 Default Settings

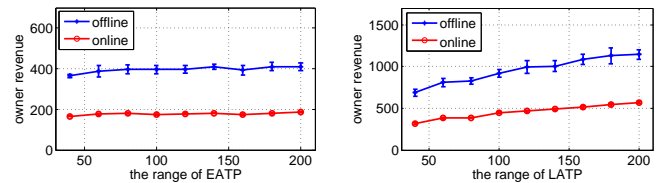
We randomly generate the sensing values of tasks, the number of users and the users’ private values. More specifically, the sensing cost per unit time (i.e.,  $\hat{d}_i$ ) of any user  $A_i$  is generated randomly from the uniform distribution  $U[0.1, 1.1]$ , and the sensing value per unit time  $u_i$  of any task  $K_i$  is also generated from the same distribution (i.e., both are bounded away from 0). The earliest available time point  $\hat{s}_i$  of any user  $A_i$  is generated randomly from  $U[0, 100]$ , whereas the length of  $A_i$ ’s available time period is generated randomly from  $U[0, 10]$ . Both the number of users  $n$  and the budget  $G$  are set to 1000; the number of tasks  $m$  is set 100; and each user selects only one task, with equal probability out of all tasks. All our simulations follow these default settings unless otherwise stated.

### 6.2 Owner Revenue

We study the owner revenue achieved by our mechanisms under different user number, budget and task number in Fig. 3. For each data point, we perform 100 simulations with random inputs and we plot the average value and standard deviation. In general, offline always works better than online. This is natural because online faces a harsher condition that a schedule has to be determined for a user upon his/her arrival. In particular, with the information on all users, offline can leverage a sorting based on  $\prec$  to optimize the performance, whereas online do not have this privilege.

In Fig. 3(a), we study the impact of the number of users on the owner revenue, by scaling the number of users from 500 to 5000 with an increment of 100 (below 1000) and of 1000 (beyond 1000). The owner revenues of both offline and online increase with the number of users. This can be explained by the reason that, as the diversities of both the users’ sensing costs and available time periods increase with the number of users, the degree of freedom in finding schedules is enlarged, which in turn results in larger revenues. The same trend is also shown in Fig. 3(b), where we fix the number of users to 1000 but scale the budget  $G$  from 200 to 2000 with an increment of 100 (before 1000) and of 500 (after 1000). This is rather straightforward to understand because a higher budget allows the algorithms to schedule users whose sensing costs are higher and hence cannot be afforded under a lower budget.

In Fig. 3(c), we fix the user number to 1000 and increase the number of tasks from 20 to 200 with an increment of 20. It can be seen that the owner revenues obtained by both our mechanisms slightly increase when the number of tasks increases. This can be explained by the reason that, when the number of tasks increases, the number of users that can perform each task tend to decrease, which results in less overlapping available time periods and higher revenue. Obviously, this effect is less direct than that from either increasing user number or budget, so the resulting improvement to the revenue is also marginal.



(a) Changing the range of EATP (b) Changing the range of LAMP

Fig. 5. The owner’s revenue vs. the range of time intervals

In Fig. 5, we study how the owner revenue is impacted by the ranges of time intervals we set in Sec. 6.1. In Fig. 5(a), the Length of Available Time Period (LAMP) of any user is generated randomly from  $U[0, 10]$ , whereas the Earliest Available Time Point (EATP) of any user is generated randomly from the uniform distribution  $U[0, x]$  with  $x$  ranging from 40 to 200 by a step of 20. In Fig. 5(b), the EATP of any user is generated randomly from  $U[0, 200]$  and the LAMP of any user is generated randomly from  $U[0, y]$  with  $y$  ranging from 40 to 200 by a step of 20. It can be seen that the owner’s revenue does not vary much when the range of EATP changes, while it increases slowly with the increment of the range of LAMP. This provides us the insight that the lengths of available time periods of the users affect the sensing revenue more significantly, as longer LAMPs provide more time slots to be allocated to the users and increase the searching space of our algorithms. Moreover, it can be seen that, in both



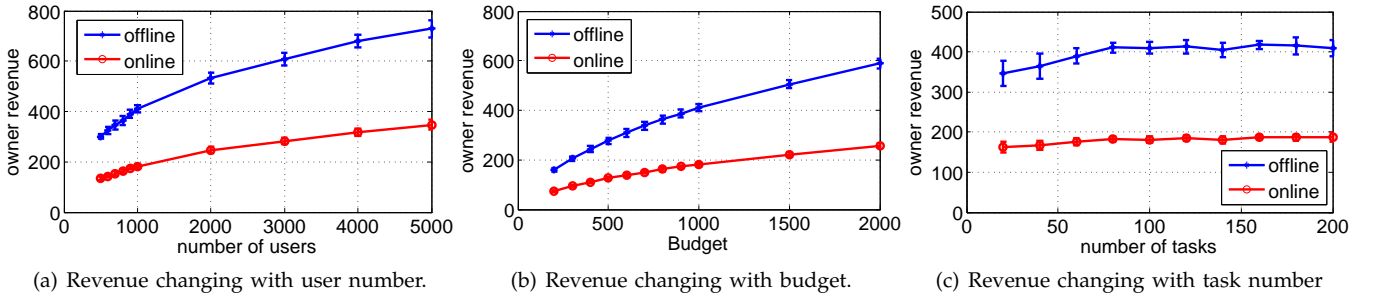


Fig. 3. The owner's revenue as functions of (a) the number of users, (b) the total budget, and (c) the number of tasks.

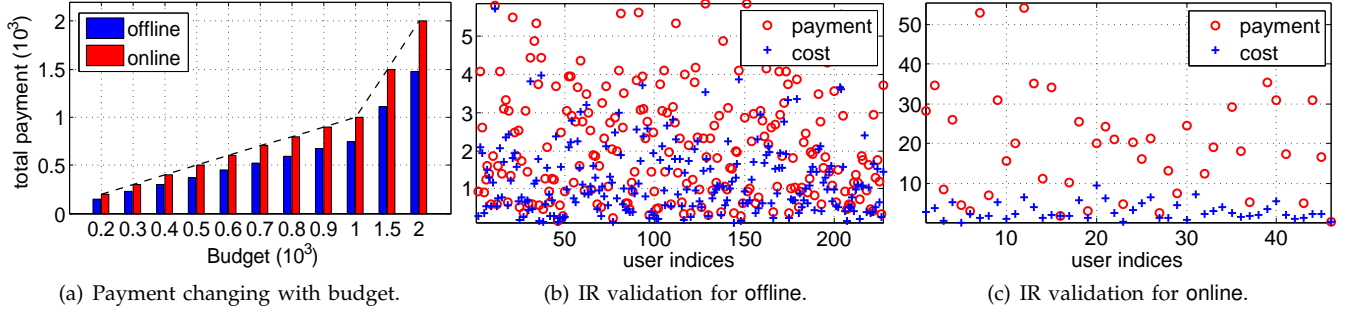


Fig. 4. Feasibility check in terms of total payment and IR.

settings of Fig. 5(a) and Fig. 5(b), offline outperforms online, which is the same with that shown in Fig. 3. This indicates that our mechanisms are robust against the changing of setup values on time intervals.

### 6.3 Solution Feasibility and Individual Rationality

We verify the feasibility and IR of the solutions output by our algorithms in this section. We first show that the total payment is always no more than the budget, then we use two examples to demonstrate that IR is also guaranteed, i.e., each user gets a payment higher than his/her cost. In Fig. 4(a), we scale the budget in the same way as Fig. 3(b), and we show the maximum total payment for each case. Apparently, the budget has never been surpassed. Again, offline is shown to be superior to online: it results in lower total payments.

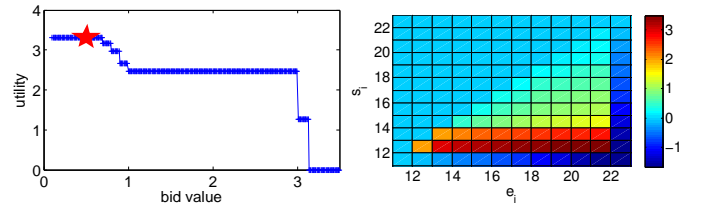
In Fig. 4(b) and 4(c), we demonstrate IR using the outputs from offline and online, respectively. We plot the sensing costs and payments only for users with non-zero payments. IR of our mechanisms can be immediately seen: a payment is always greater than the corresponding cost. We can also see that more users are assigned non-empty schedules by offline, which, to some extent, explains the observation made for Fig. 3 that offline always achieves a higher revenue than online.

### 6.4 Truthfulness

We verify the truthfulness of both offline and online by arbitrarily picking up a few users and checking their utilities under different bidding values.

We first study the truthfulness of offline by Fig. 6. We arbitrarily pick a user  $A_i$  whose true values are  $\hat{d}_i = 0.5$ ,  $\hat{s}_i = 12$  and  $\hat{e}_i = 22$ , then we change  $A_i$ 's

bid (with other users' bids fixed) to see how  $A_i$ 's utility changes. Since we cannot draw a 4-dimensional chart here, we show the results by two figures. In Fig. 6(a), the bid of  $A_i$ 's available time period is fixed to  $[12, 22]$ , and we scale  $A_i$ 's bid on his sensing cost from 0.1 to 3.3 with an increment of 0.1. Indeed, bidding the true sensing cost (shown by the red pentagram) allows  $A_i$  to maximize his/her utility. In Fig. 6(b), we fix  $A_i$ 's bid on his sensing cost to 0.5, but varies  $A_i$ 's bid on his earliest and latest available time points. Again,  $A_i$ 's utility is maximized when he/she bids his/her true value  $[12, 22]$ . These demonstrate that  $A_i$  has no incentive to deviate from bidding his/her true values.



(a) Truthfulness on the sensing cost. (b) Truthfulness on the available time.

Fig. 6. Truthfulness validation for offline.

Similarly, we study the truthfulness of online by Fig. 7. In Fig. 7(a), we pick an arbitrary user whose true sensing cost is 0.3 and who is assigned an empty schedule by online. Then we scale this user's bid on his sensing cost from 0.01 to 0.4 with an increment of 0.01. The user indeed achieves his/her maximum utility 0 by bidding his true sensing cost 0.3. In Fig. 7(b), we pick another user (in another simulation) whose true sensing cost is 0.4 and who is assigned a non-empty schedule by

online. Then we scale this user’s bid from 0.01 to 10 with an increment of 0.01. Again, the result shows that the user’s utility is maximized when he/she bids his/her true sensing cost.

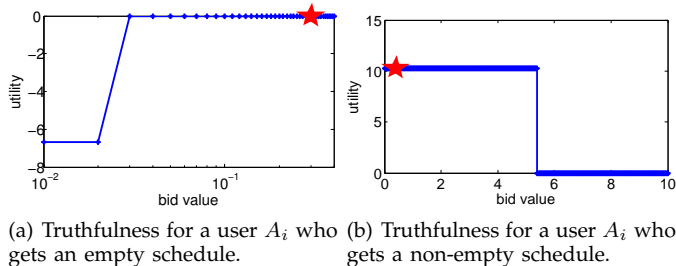


Fig. 7. Truthfulness validation for online.

### 6.5 Comparing with related work

We compare our mechanisms with a recent work [10], where truthful mechanisms are also proposed for crowdsensing. However, as the problem studied in [10] is very different from ours and no time-scheduling problem is considered there (as we will discuss in Sec. 7), it is very hard to adapt the mechanisms proposed in [10] to fit our case perfectly. Therefore, we have to adapt the “OMG” mechanism proposed in [10] to a mechanism “OMG\_A” to solve our problem, where OMG\_A obtains the same revenue with that of OMG, but does not guarantee truthfulness as we do. More specifically, OMG\_A accepts the same input as our mechanisms, but once a user  $A_i$  is selected by OMG\_A, all the available time slots of  $A_i$  are assigned to  $A_i$  for sensing and the sensing cost of  $A_i$  is calculated as the whole sensing cost of  $A_i$ ’s available time period. This way of adapting OMG to our problem is the direct consequence of OMG’s lack of time-scheduling ability.

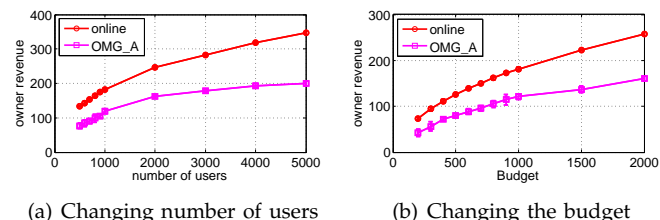


Fig. 8. Comparing with OMG

In Fig. 8(a), we compare the performance of OMG\_A with online when the number of users increases in the same manner with that in Fig. 3(a); In Fig. 8(b), OMG\_A and online are compared when the budget increases in the same manner with that in Fig. 3(b). It can be seen that online outperforms OMG\_A in both cases. This can be understood by the fact that the OMG mechanism was not designed purposely for the time-scheduling problem studied in our paper, hence it is hard to get ideal performance by adapting OMG to our problem.

## 7 RELATED WORK

Mobile crowdsensing involves using (human carried) smartphones to gather data in a much larger scale than what can be done in conventional ways, either through autonomous phone sensing or by further demanding active human participation [1], [2]. While the developments on mobile crowdsensing applications are plentiful, only a few proposals have started on studying how to incentivize participation to such applications until very recently [3]–[7], [16].

Yang *et al.* suggest both a platform-centric model and a user-centric model for mobile sensing in their seminal work [6]. They also use a Stackelberg game to design incentive mechanisms for the platform-centric model, and use auction theory to design truthful mechanisms for the user-centric model. Their mechanisms can simultaneously achieve computational efficiency, individual rationality, profitability and truthfulness. Nevertheless, the truthful auction mechanisms provided in [6] are only for single-parameter users and only run in an offline manner. Moreover, no theoretical performance ratios are provided for them in [6]. Duan *et al.* [5] have proposed incentive mechanisms to motivate collaboration in mobile crowdsensing based on Stackelberg games and contract theory. A notable strength of their work is that their methods are appropriate for both data acquisition and distributed computing applications. However, the mechanisms in [5] require either the complete information or the prior distributions of users’ private types, hence are not prior-free mechanisms as those in our work. Other papers [3], [4], [7] have also proposed algorithms and conducted extensive experiments to study various issues in mobile crowdsensing such as pricing, coverage, and privacy. However, these proposals are either not based on a game theoretical perspective or have not considered important game-theoretic issues such as truthfulness and IR. Most importantly, none of the work in [3]–[7] has considered the special time scheduling problem arising from the mobile crowdsensing paradigm, hence their problem definitions are totally different from ours.

Very recently, two proposals [10], [11] have performed deep investigations on the task allocation problem in mobile crowdsensing. They proposed new incentive mechanisms using online-auction theory [8], [17] and economic properties including truthfulness are achieved by the proposed mechanisms. Nevertheless, similar to the work in [3]–[7], the problems studied in [10], [11] are intrinsically different from ours, as neither of them considered the sensing-time scheduling problem in crowdsensing, which is exactly the focus of our paper. Actually, both [10] and [11] assume that any sensing task can be done instantly, hence the users’ sensing costs and payments in their proposed mechanisms are irrelevant to the sensing time. Besides, unlimited budget is assumed in [11]. Due to these critical discrepancies on models and problem formulations, the mechanisms proposed in [10], [11] cannot be adapted to our case.



S. Since the Partition problem is NP-complete, the MCS problem is NP-hard.  $\square$

**Proof of Theorem 2:** As the users' strategic behaviours are not considered here, it can be easily seen by line 13 in **Algorithm 1** that any user can always get a payment no less than his sensing cost. Hence we only need to prove that the total amount paid to the users is no more than the budget  $G$ . For any  $1 \leq i \leq h$  we have

$$\begin{aligned} R(\vec{y}_i) &= R(\vec{y}_{i-1}) + |y_{n_i}| \mu_{n_i} \\ &\leq R(\vec{y}_{i-1}) + \mu_{n_i} \left( \frac{G}{2d_{n_i}} - \frac{R(\vec{y}_{i-1})}{\mu_{n_i}} \right) \\ &= \mu_{n_i} G / (2d_{n_i}), \end{aligned}$$

hence we get

$$\mu_{n_i} / d_{n_i} \geq 2R(\vec{y}_i) / G, \quad (5)$$

and

$$\mu_{n_1} / d_{n_1} \geq \mu_{n_2} / d_{n_2} \geq \dots \geq \mu_{n_h} / d_{n_h} \geq 2R(\vec{y}) / G.$$

So the total amount paid to the users is

$$\sum_{1 \leq i \leq h} d_{n_i} \cdot |y_{n_i}| \leq \sum_{1 \leq i \leq h} \mu_{n_i} \cdot |y_{n_i}| \cdot \frac{G}{2R(\vec{y})} = \frac{G}{2}. \quad (6)$$

Therefore, **Algorithm 1** yields a feasible solution.  $\square$

**Proof of Theorem 3:** Suppose that **Algorithm 1** has  $l$  effective iterations if we replace line 7 by

$$q \leftarrow \min \left\{ |Z_j|, \frac{G - \sum_{i=1}^n d_i * |w_i(\vec{b})|}{d_j} \right\}, \quad (7)$$

and let  $\vec{x}_i$  be the current vector  $\vec{w}(\vec{b})$  after the  $i$ th effective iteration ( $1 \leq i \leq l$ ) is executed in this case. Clearly,  $h \leq l$ , and the user scheduled in the  $i$ th effective iteration under this case can also be denoted by  $A_{n_i} : 1 \leq i \leq l$ . Let  $R'(\vec{x}_i) = \sum_{j=1}^i \mu_{n_j} |x_{n_j}|$ .

Let  $\mathcal{X}_0 = \emptyset$  and  $\mathcal{X}_i = \{A_{n_1}, A_{n_2}, \dots, A_{n_i}\}$  for any  $1 \leq i \leq l$ . From **Algorithm 1** we know that, for any  $1 \leq j < l$  and any  $1 \leq i \leq j$ ,  $T_{n_i}$  must be covered by  $\vec{x}_j$ , i.e.,  $T_{n_i} \subseteq \bigcup_{1 \leq \ell \leq j \wedge \kappa_{n_\ell} = \kappa_{n_i}} f_{n_\ell}$  where  $\vec{f} = \vec{x}_j$ . Therefore, we have

$$\begin{aligned} R(\vec{w}^*) - R'(\vec{x}_j) &\leq \sum_{i: A_i \in \mathcal{A} \setminus \mathcal{X}_j} |w_i^*| \mu_i \\ &= \sum_{i: A_i \in \mathcal{A} \setminus \mathcal{X}_j} |w_i^*| d_i \cdot \frac{\mu_i}{d_i} \leq G \cdot \frac{\mu_{n_{j+1}}}{d_{n_{j+1}}} \quad (8) \\ &= G \cdot \frac{|x_{n_{j+1}}| \mu_{n_{j+1}}}{|x_{n_{j+1}}| d_{n_{j+1}}} = G \cdot \frac{R'(\vec{x}_{j+1}) - R'(\vec{x}_j)}{|x_{n_{j+1}}| d_{n_{j+1}}} \quad (9) \end{aligned}$$

where (8) holds because of the greedy selection rule in line 5. This yields

$$\begin{aligned} R'(\vec{x}_{j+1}) &\geq \frac{|x_{n_{j+1}}| d_{n_{j+1}}}{G} R(\vec{w}^*) \\ &\quad + \left( 1 - \frac{|x_{n_{j+1}}| d_{n_{j+1}}}{G} \right) R'(\vec{x}_j) \quad (10) \end{aligned}$$

Note that equation (9) and (10) also hold for  $j = 0$  since  $R'(\vec{x}_0) = 0$ . Therefore, when  $k = 1$ , we have:

$$R'(\vec{x}_k) \geq \left[ 1 - \prod_{i=1}^k \left( 1 - \frac{|x_{n_i}| d_{n_i}}{G} \right) \right] R(\vec{w}^*) \quad (11)$$

By induction and using equation (10), for any  $1 < k \leq l$ , we also have

$$\begin{aligned} R'(\vec{x}_k) &\geq \frac{|x_{n_k}| d_{n_k}}{G} R(\vec{w}^*) \\ &\quad + \left( 1 - \frac{|x_{n_k}| d_{n_k}}{G} \right) \left[ 1 - \prod_{i=1}^{k-1} \left( 1 - \frac{|x_{n_i}| d_{n_i}}{G} \right) \right] R(\vec{w}^*) \\ &= \left[ 1 - \prod_{i=1}^k \left( 1 - \frac{|x_{n_i}| d_{n_i}}{G} \right) \right] R(\vec{w}^*), \end{aligned}$$

which means that equation (11) holds for any  $1 \leq k \leq l$ .

Now we assume that  $\sum_{i=1}^l |x_{n_i}| d_{n_i} = G$ . In this case, using equation (11) we can get:

$$\begin{aligned} R'(\vec{x}_l) &\geq \left[ 1 - \prod_{j=1}^l \left( 1 - \frac{|x_{n_j}| d_{n_j}}{G} \right) \right] R(\vec{w}^*) \\ &= \left[ 1 - \prod_{j=1}^l \left( 1 - \frac{|x_{n_j}| d_{n_j}}{\sum_{i=1}^l |x_{n_i}| d_{n_i}} \right) \right] R(\vec{w}^*) \\ &\geq (1 - (1 - 1/l)^l) R(\vec{w}^*) \\ &\geq (1 - 1/e) R(\vec{w}^*) \quad (12) \end{aligned}$$

On the other side, if  $\sum_{i=1}^l |x_{n_i}| d_{n_i} < G$ , then we must have  $T_j \subseteq \bigcup_{1 \leq i \leq l} x_{n_i}$  for any  $j \notin \{n_1, n_2, \dots, n_l\}$ , because otherwise  $|x_j| \neq 0$ , which is a contradiction. This implies that  $R'(\vec{x}_l) \geq R(\vec{w}^*)$ . Consequently, we know that equation (12) always holds.

It can be seen that  $|y_{n_i}| = |x_{n_i}|$  for any  $1 \leq i \leq h-1$  and  $|y_{n_h}| \leq |x_{n_h}|$ . From line 10 we know  $|y_{n_h}| = \lfloor \frac{G}{2d_{n_h}} - \frac{R(\vec{y}_{h-1})}{\mu_{n_h}} \rfloor$ . Hence we get

$$|x_{n_h}| \geq \frac{G}{2d_{n_h}} - \frac{R(\vec{y}_{h-1})}{\mu_{n_h}} - 1 = \frac{G}{2d_{n_h}} - \frac{R'(\vec{x}_{h-1})}{\mu_{n_h}} - 1$$

This yields

$$R'(\vec{x}_h) = R'(\vec{x}_{h-1}) + |x_{n_h}| \mu_{n_h} \geq G \cdot \mu_{n_h} / (2d_{n_h}) - \mu_{n_h}$$

Therefore, for any  $h \leq i \leq l$  we have

$$\mu_{n_i} / d_{n_i} \leq \mu_{n_h} / d_{n_h} \leq 2(R'(\vec{x}_h) + \mu_{n_h}) / G \quad (13)$$

Since  $R'(\vec{x}_{h-1}) \leq R(\vec{y})$ , using equation (13) we get:

$$\begin{aligned} R'(\vec{x}_l) &= R'(\vec{x}_{h-1}) + \sum_{i=h}^l \mu_{n_i} |x_{n_i}| \\ &\leq R(\vec{y}) + \frac{2(R'(\vec{x}_h) + \mu_{n_h})}{G} \cdot \sum_{i=h}^l d_{n_i} |x_{n_i}| \\ &\leq R(\vec{y}) + 2(R'(\vec{x}_{h-1}) + \mu_{n_h} |x_{n_h}|) + 2\mu_{n_h} \\ &\leq 3R(\vec{y}) + 4\mu_{n_h} |x_{n_h}| \\ &\leq 3R(\vec{y}) + (1 - 1/e - 4\epsilon) R(\vec{w}^*) \end{aligned}$$

Combing this with equation (12), the theorem follows.  $\square$

**Proof of Lemma 1:** Because  $\mu_i/d'_i \leq \mu_i/d_i$ , bidding  $b'_i$  can only postpone the schedule assignment for  $A_i$  according to line 5 of **Algorithm 1**. Hence the length of the time period scheduled for  $A_i$  can only decrease when  $A_i$  bids  $b'_i$ , due to line 7 of **Algorithm 1**.  $\square$

**Proof of Lemma 2:** Let  $\widehat{Z}$  and  $\widehat{Z}'$  be the uncovered available time period of  $A_i$  when  $A_i$  is scheduled by **Algorithm 1** with the input bids being  $(b_i, b_{-i})$  and  $(b'_i, b_{-i})$ , respectively. As  $[s'_i, e'_i] \subseteq [s_i, e_i]$ , we must have  $\widehat{Z}' \subseteq \widehat{Z}$ . Hence the lemma follows due to lines 7 and 10 of **Algorithm 1**.  $\square$

**Proof of Lemma 3:** Suppose that  $j = n_c : 1 \leq c \leq h$  (hence  $j \in \{n_1, n_2, \dots, n_h\}$ ) and  $d_j > \frac{\mu_j \cdot G}{R(\vec{y})}$  by contradiction, we have

$$\begin{aligned} R(\vec{y}) - R(\vec{y}_c) &\leq \sum_{c \leq i \leq h} |y_{n_i}| \cdot d_{n_i} \cdot (\mu_{n_i}/d_{n_i}) \\ &\leq \frac{\mu_j}{d_j} \cdot \sum_{1 \leq i \leq h} |y_{n_i}| \cdot d_{n_i} \leq \frac{G}{2} \cdot \frac{\mu_j}{d_j} \quad (14) \\ &< R(\vec{y})/2, \end{aligned}$$

where (14) holds due to (6). Therefore  $R(\vec{y}) < 2R(\vec{y}_c)$ . On the other side, (5) suggests  $\mu_j/d_j \geq 2R(\vec{y}_c)/G$ . Combining these inequalities yields  $d_j < \frac{\mu_j \cdot G}{R(\vec{y})}$ ; a contradiction.  $\square$

**Proof of Theorem 5:** It is easy to see that lines 8-11 of **Algorithm 2** can output a feasible solution satisfying IR to the MCS problem. The output of lines 3-6 satisfies IR according to **Theorem 4**. Hence we only need to prove that  $\sum_{j \in \mathcal{W}} p_j(\vec{b}) \leq G$  for  $\mathcal{W} = \{n_1, n_2, \dots, n_h\}$ . According to **Lemma 3**, no user  $A_j : j \in \mathcal{W}$  can bid  $(d_j, s_j, e_j)$  with  $d_j > \frac{\mu_j \cdot G}{R(\vec{y})}$ , because otherwise he/she will get an empty schedule. Therefore, using **Theorem 4** and **Lemma 1** we can get

$$\begin{aligned} &p_j(b_j, b_{-j}) \\ &= d_j \cdot |y_j(b_j, b_{-j})| + \int_{d_j}^{\infty} |y_j((v, s_j, e_j), b_{-j})| dv \\ &= d_j \cdot |y_j(b_j, b_{-j})| + \int_{d_j}^{\frac{\mu_j \cdot G}{R(\vec{y})}} |y_j((v, s_j, e_j), b_{-j})| dv \\ &\leq d_j \cdot |y_j(b_j, b_{-j})| + \left( \frac{\mu_j \cdot G}{R(\vec{y})} - d_j \right) \cdot |y_j(b_j, b_{-j})| \\ &= (\mu_j \cdot G / R(\vec{y})) \cdot |y_j(b_j, b_{-j})|. \end{aligned}$$

Given that  $R(\vec{y}) = \sum_{j \in \mathcal{W}} \mu_j \cdot |y_j(b_j, b_{-j})|$ , we can prove  $\sum_{j \in \mathcal{W}} p_j(\vec{b}) \leq G$  by summing up  $p_j(b_j, b_{-j})$  for all  $j \in \mathcal{W}$ , hence the theorem follows.  $\square$

**Proof of Theorem 7:** For any  $\epsilon \in (0, \frac{e-1}{4e})$ , if (1) is satisfied, then the mechanism in **Algorithm 2** has a revenue of at least  $\frac{4}{3}\epsilon \cdot R(\vec{y}^*)$  with probability of  $\frac{1}{2}$ ; if (1) is not satisfied, then we have:

$$\mu_j \geq \max_{i: d_i \leq G} (\mu_i \cdot |T_i|) / \lambda \geq \left( \frac{e-1}{4\lambda e} - \frac{\epsilon}{\lambda} \right) R(\vec{y}^*),$$

hence the mechanism has a revenue of at least  $(\frac{e-1}{4\lambda e} - \frac{\epsilon}{\lambda}) \cdot R(\vec{y}^*)$  with probability of  $\frac{1}{2}$ . Therefore, the overall ap-

proximation ratio of the mechanism is  $\mathcal{O}(1)$ . For example, if we set  $\epsilon = \frac{3}{28}(1 - 1/e)$ , then the expected revenue of the mechanism is at least  $\frac{1}{\lambda}(1 - 1/e) \cdot R(\vec{y}^*)$ .  $\square$

**Proof of Theorem 8:** Line 3 of **Algorithm 2** calls **Algorithm 1** that has a time complexity of  $\mathcal{O}(n^2)$  due to the sorting of the users. Line 5 is iterated at most  $n$  times and each calculates the payment to one user by calling **Algorithm 3** that has a time complexity of  $\mathcal{O}(n)$ . The time complexity of lines 8-11 in **Algorithm 2** is  $\mathcal{O}(n)$ . Consequently, the overall time complexity of **Algorithm 2** is  $\mathcal{O}(n^2)$ .  $\square$

**Proof of Theorem 10:** Similar to the secretary algorithm [15], we can prove that the user  $A_k$  is selected with probability of at least  $1/e - 1/3$ , where  $k = \arg \max_{i: d_i \leq G} \mu_i$ , as far as there are more than two users (i.e.,  $n \geq 3$ ). Hence **Algorithm 4** has a constant competitive ratio of  $1/(150\lambda)$  with probability of at least  $1/e - 1/3$ .  $\square$

**Proof of Lemma 4:** Let  $\{Y_1, Y_2, \dots, Y_n\}$  be a set of independent random variables such that  $Y_i = |y_i^*| \cdot \mu_i$  if  $i \in \{\sigma_1, \sigma_2, \dots, \sigma_\xi\}$ , and  $Y_i = 0$  otherwise. Clearly,  $Y_i \leq \Lambda$  for any  $1 \leq i \leq n$ . Let  $Y = \sum_{i=1}^n Y_i$ . Hence,  $\Delta_1 = Y$  and  $\Delta_2 = R(\vec{y}^*) - Y$ . For any  $1 \leq i \leq n$ , we have

$$\begin{aligned} &\text{Prob}(i \in \{\sigma_1, \sigma_2, \dots, \sigma_\xi\}) \\ &= \sum_{j=1}^n \text{Prob}(\sigma_j = i) \cdot \text{Prob}(\xi \geq j) = \frac{1}{n} \mathbb{E}(\xi) = \frac{1}{2}. \end{aligned}$$

So we know  $\mathbb{E}(Y) = \sum_{i=1}^n \mathbb{E}(Y_i) = \frac{1}{2} \sum_{i=1}^n |y_i^*| \cdot \mu_i = \frac{R(\vec{y}^*)}{2}$ . According to the Chernoff bound, we get

$$\begin{aligned} \text{Prob}\left(\Delta_1 \leq \frac{R(\vec{y}^*)}{3}\right) &= \text{Prob}\left(Y \leq \left(1 - \frac{1}{3}\right) \mathbb{E}(Y)\right) \\ &\leq e^{-\frac{\frac{1}{3} \cdot \mathbb{E}(Y)}{2\Lambda}} = e^{-\frac{R(\vec{y}^*)}{36\Lambda}} \leq e^{-\frac{150}{36}} \leq 0.016, \end{aligned}$$

and

$$\begin{aligned} \text{Prob}\left(\Delta_2 \leq \frac{R(\vec{y}^*)}{4}\right) &= \text{Prob}\left(Y \geq \left(1 + \frac{1}{2}\right) \mathbb{E}(Y)\right) \\ &\leq \left(\frac{e^{\frac{1}{2}}}{\left(1 + \frac{1}{2}\right)^{\left(1 + \frac{1}{2}\right)}}\right)^{\frac{\mathbb{E}(Y)}{\Lambda}} \leq (0.9)^{\frac{R(\vec{y}^*)}{2\Lambda}} \leq 0.9^{75} \leq 0.001. \end{aligned}$$

By the union bound, we know

$$\text{Prob}\{\Delta_1 \geq R(\vec{y}^*)/3 \wedge \Delta_2 \geq R(\vec{y}^*)/4\} \geq 0.983.$$

So the lemma follows.  $\square$

**Proof of Lemma 5:** Let  $opt_1$  be the revenue of the optimal solution for the first arrived  $\xi$  users  $\{A_{\sigma_1}, \dots, A_{\sigma_\xi}\}$ . Using **Theorem 3** with  $\epsilon = \frac{e-1}{4e} - \frac{1}{150}$ , we get

$$R(\vec{r}) \geq \frac{4}{3} \cdot \left( \frac{e-1}{4e} - \frac{1}{150} \right) \cdot opt_1 \geq \frac{1}{5} \cdot opt_1.$$

As  $opt_1 \geq \Delta_1$ , we get  $R(\vec{r}) \geq \Delta_1/5$ . On the other hand,  $R(\vec{r}) \leq opt_1 \leq R(\vec{y}^*)$ , hence the lemma follows.  $\square$

**Proof of Lemma 6:** Let  $O_1 = \{\sigma_i | \xi + 1 \leq i \leq n \wedge y_{\sigma_i}^* \notin \bigcup_{j: \kappa_j = \kappa_{\sigma_i}} y_j\}$  and  $O_2 = \{\sigma_i | \xi + 1 \leq i \leq$

$n \wedge y_{\sigma_i}^* \subseteq \bigcup_{j:\kappa_j=\kappa_{\sigma_i}} y_j\}$ . Since  $\sum_{i \in O_2} |y_i^*| \cdot \mu_i \leq R(\bar{y})$ , we have

$$\begin{aligned} \Delta_2 - R(\bar{y}) &= \sum_{i \in O_1} |y_i^*| \cdot \mu_i + \sum_{i \in O_2} |y_i^*| \cdot \mu_i - R(\bar{y}) \\ &\leq \sum_{i \in O_1} |y_i^*| \cdot \mu_i. \end{aligned}$$

For any  $i \in O_1$ , we must have  $|y_i^*| \neq 0$ ,  $|y_i| = 0$ , and  $|F_i| \geq 1$ . So we can discuss line 13 as follows:

Case 1: Suppose that  $d_i > \eta = 5G \cdot \mu_i / R(\bar{r})$  for any  $i \in O_1$ . Using *Lemma 5* we have:

$$\begin{aligned} \Delta_2 - R(\bar{y}) &\leq \sum_{i \in O_1} |y_i^*| \cdot d_i \cdot R(\bar{r}) / (5G) \\ &\leq R(\bar{r}) / 5 \leq R(\bar{y}^*) / 5. \end{aligned}$$

As  $R(\bar{y}^*) \leq 4\Delta_2$  according to *Lemma 4*, we have

$$R(\bar{y}) \geq \Delta_2 - R(\bar{y}^*) / 5 \geq R(\bar{y}^*) / 20.$$

Case 2: Suppose that there exists  $i \in O_1$  such that  $d_i \leq 5G \cdot \mu_i / R(\bar{r})$  but  $\eta|F_i| = 5|F_i| \cdot G \cdot \mu_i / R(\bar{r}) > M$ . In this case, using *Lemma 4* and *Lemma 5* we get:

$$\begin{aligned} M &\leq 5|F_i| \cdot G \cdot \mu_i / R(\bar{r}) \leq 5G \cdot \Lambda / R(\bar{r}) \\ &\leq 5G \cdot \frac{R(\bar{y}^*)}{150} / \left( \frac{\Delta_1}{5} \right) \leq \frac{G}{2}, \end{aligned} \quad (15)$$

hence

$$\begin{aligned} \frac{G}{2} &\leq \sum_{i=\xi+1}^n p_{\sigma_i} \leq \sum_{i=\xi+1}^n |y_{\sigma_i}| \cdot 5G \cdot \mu_{\sigma_i} / R(\bar{r}) \\ &= 5G \cdot R(\bar{y}) / R(\bar{r}) \leq 25G \cdot R(\bar{y}) / \Delta_1 \\ &\leq 75G \cdot R(\bar{y}) / R(\bar{y}^*), \end{aligned} \quad (16)$$

which yields  $R(\bar{y}) \geq \frac{1}{150} R(\bar{y}^*)$ . Therefore, lines 3-16 of *Algorithm 5* has a competitive ratio of 1/150 with probability of at least 0.983.  $\square$

*Proof of Theorem 12:* Similar to the proof of *Theorem 7*, it can be easily proven that *Algorithm 5* has an  $\mathcal{O}(1)$  competitive ratio based on *Lemma 6* and *Theorem 10*. *Algorithm 5* has its running time predominantly spent on line 10, which has a  $\mathcal{O}(n^2)$  worst-case time complexity.  $\square$



**Kai Han** received the B.S. and Ph.D. degrees in computer science from the University of Science and Technology of China, Hefei, China, in 1997 and 2004, respectively.

He is currently a Professor at the School of Computer Science and Technology, University of Science and Technology of China, China. His research interests include wireless ad hoc and sensor networks, mobile and cloud computing, combinatorial and stochastic optimization, algorithmic game theory, as well as machine learning. He is a Member of both IEEE and ACM.



**Chi Zhang** received the B.S. degree in computer science from Zhejiang University, Hangzhou, China, and is currently pursuing the Ph.D. degree in computer science at Nanyang Technological University, Singapore. His research interests include wireless sensor networks and social networks.



**Jun Luo** got his PhD degree in computer science from EPFL (Swiss Federal Institute of Technology in Lausanne), Lausanne, Switzerland, in 2006. From 2006 to 2008, he has worked as a post-doctoral research fellow in the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. In 2008, he joined the faculty of the School of Computer Engineering, Nanyang Technological University in Singapore, where he is currently an Associate Professor. His research interests

include wireless networking, distributed systems, multimedia protocols, network modeling and performance analysis, applied operations research, as well as network security. He is a Member of both IEEE and ACM.



**Menglan Hu** received the B.E. degree from Huazhong University of Science and Technology, Wuhan, China, in 2007, and the Ph.D. degree from the National University of Singapore, Singapore, in 2012. He is currently a research fellow at the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include cloud computing, parallel and distributed systems, as well as scheduling and resource management.



**Bharadwaj Veeravalli** received his BSc in Physics, from Madurai-Kamaraj University, India in 1987, Masters in Electrical Communication Engineering from Indian Institute of Science, Bangalore, India in 1991 and PhD from Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India in 1994. He did his post-doctoral research in Concordia University, Montreal, Canada, in 1996. He is currently with the Department of Electrical and Computer Engineering at the National University of Singapore, Singapore, as a tenured Associate Professor. His research interests include CloudGridCluster Computing, Scheduling in Parallel and Distributed Systems, Bioinformatics & Computational Biology, and Multimedia Computing. He is one of the earliest researchers in Divisible Load Theory (DLT). He had secured several externally funded projects and published over 120 papers in high-quality journals and conferences. He has co-authored three research monographs in the areas of PDS, Distributed Databases, and Networked Multimedia Systems, in 1996, 2003, and 2005, respectively. He is currently serving the Editorial Board of IEEE Transactions on Computers and IEEE Transactions on Cloud Computing, and Cluster Computing, as an Associate Editor. More information can be found in <http://cnl-ece.nus.edu.sg/elebv>. He is a senior member of the IEEE and the IEEE computer society.

He is currently a Professor at the School of Computer Science and Technology, University of Science and Technology of China, China. His research interests include wireless ad hoc and sensor networks, mobile and cloud computing, combinatorial and stochastic optimization, algorithmic game theory, as well as machine learning. He is a Member of both IEEE and ACM.