Fault Tolerant Mechanism Design for Time Coverage in Crowdsensing System

Yangsu Liu, Zhenzhe Zheng, Fan Wu, Xiaofeng Gao and Guihai Chen
Shanghai Key Laboratory of Scalable Computing and Systems
Department of Computer Science and Engineering
Shanghai Jiao Tong University, China

***Abstract*—Last decade has witnessed the explosion in the number of smart mobile phones and other smart devices equipped with powerful sensors. Therefore, various crowdsensing applications which recruit people to complete sensing tasks have sprung up. Designing incentive mechanism plays an indispensable role in crowdsensing network. However, most existing works about incentive mechanism base on the assumption that agents will complete the allocated sensing tasks without any problem. However, when we take the failures of agents into consideration, most existing incentive mechanisms become invalid.**

Considering a more practical scenario, we suppose that there is a crucial sensing task which needs to remain high enough probability of success completion for a certain period of time and each alternative agent has a certain probability to cover a certain period successfully. To ensure the fault tolerance of the crowdsensing system, we propose two novel incentive mechanisms, single slot coverage (SSC) mechanism and continuous coverage (CC) mechanism, for different problem models, respectively. In our mechanisms, agents' probabilities of success, costs of completing task, start time and end time are all private information. Our objective is to minimize the total costs of selected agents, while ensuring the task is fully covered with a high enough probability over a certain period. Our work presents detailed proofs of the computational efficiency, truthfulness and individual rationality. Besides, we implement extensive simulation to evaluate proposed mechanisms, which validates the properties of our mechanisms.

I. Introduction

Following the explosive growth in the number of smart phones, various kind of embedded sensors on mobile phones are available for collecting information, such as GPS, accelerometers, digital compasses, microphones, and cameras. Furthermore, smart phones also integrate communication module, storage module and computation module. It indicates that ubiquitous smart phones are promising to play the role of sensors in classic wireless sensor network(WSN). Since the enormous potential of mobile phones, mobile crowdsensing, as a novel sensing paradigm, has attracted great attention [1]. Various applications have been developed in wide fields, such as transportation planning, environment monitoring, localization, healthcare and so on [2]–[9].

This work was supported in part by the State Key Development Program for Basic Research of China (973 project 2014CB340303), in part by China NSF grant 61672348, 61672353, 61422208, and 61472252, in part by Shanghai Science and Technology fund 15220721300, in part by the CCF-Tencent Open Research Fund and in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars. The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government.

However, aforementioned applications, which recruit agents to complete sensing tasks, need sufficient number of participants. Participants consume some resources to complete the sensing tasks and bear the risk of disclosure of privacy. Few agents prefer to participate in sensing unless they are motivated enough. Therefore, incentive mechanism is essentially important in crowdsensing, which is is covered with a number of works in the survey [10]. Nevertheless, most existing incentive mechanisms are proposed on the basis of assumption that agents can complete allocated tasks without any problem. But in pracice, failures of sensing tasks are common. For example, we suppose that there is a task requiring people to monitor noise in a target area continually. Due to the attenuation of acoustical signal in propagation, the probability of monitoring noise correctly decreases with increasing distance [11]. In other sensing tasks, agents' departure halfway, low-quality data and the interference of environment also lead to failure of the task.

In this work, we consider a continuous sensing task that needs a guaranteed success probability for a period of time. Agents prefer to cover the task over a certain period of time and they maintain a certain probability of success (PoS) over the sensing process. We assume that the agents could estimate their own probability of successful sensing according to the proposed route, location, and network environment. Owing to the execution uncertainty of agents, the task needs to recruit sufficient number of agents to ensure the success rate of task satisfying the requirement of task, such that the crowdsensing system is fault tolerant. We aim at minimizing the total costs of participants while guaranteeing a threshold probability of success of the task all the time. Besides, the mechanism should persuade agents to report their private infomation truthfully, *i.e.*, the mechanism should be truthful and individually rational. Since agents are always rational and self-interested, they will try to misreport their type to gain more benefits.

In our problem model, agents could misreport their information on multi-dimensions, such as start time, end time, PoS and cost. Thus, we need to design a multi-parameter mechanism. Unfortunately, the multi-parameter mechanism design problem has no general solution yet. Thus we resort to the cost verification. Furthermore, the traditional mechanism design based on VCG mechanism [12] is invalid. This is because the problem aiming at minimizing the total costs of

978-1-5090-8468-7/17/$31.00

participants and satisfying PoS requirement simultaneously is NP-hard, and thus is computational intractable. Even if VCG mechanism can be used in NP-hard promble [13], it is not applicable to the model. Because the pricing scheme in VCG mechanism ignores the interdependencies between the probability of success and agents' valuations.

Considering these challenges, we propose two different models: single slot coverage model and continuous coverage model, where agents can select a single slot and a continuous period to cover, respectively. Due to the cost verification, We suppose that agents cannot cheat on their costs, such that the multi-dimensional mechanism design problem reduces to single-dimensional mechanism design problem. We design two incentive mechanisms on the basis of reverse auction, which both adopt greedy approximation algorithms for the agent selection. The pricing scheme is draw on the execution contingent pricing scheme in [14], which pays the agents depending on whether they complete their tasks successfully. We prove that our mechanisms both have good economical properties, such as truthfulness, individual rationality. Computation efficiency and guaranteed approximation ratio of our mechanisms are also proved theoretically.

We highlight the contribution of our work as follows:

- We formulate the problem of designing incentive mechanisms for crowdsensing based on reverse aution considering the failures of agents. We propose a crucial task needed to be covered with a certain PoS for a period of time. The goal of the incentive mechanism is to minimize the total cost of agents and ensure task to be completed with higher PoS than requirement meanwhile.
- We design SSC mechanism for the single slot coverage model. We propose a 2-approximation algorithm to for the winner selection algorithm with a greedy manner.
- We design CC mechanism for the continuous coverage model. We propose a greedy algorithm to select the agent iteratively. The analysis shows that the approximation ration of the winner selection is guaranteed .
- We extended the execution contingent pricing scheme to tackle the execution uncertainty of agents. We provide the detailed proof that the pricing scheme guarantees the truthfullness of our mechanisms.
- Extensive evaluation has validated the properties of proposed mechanisms, which is based on a real data set of CitiBike data [15] in New York.

The remainder of this paper is organized as follows. We discuss related work in Section II. Two problem models are formulated in Section III. We then design two different mechanisms for both models in Section III. We evluate our mechanisms and show the results in Section V and conclude this paper in Section VI.

II. Related Work

The problem of incentive mechanism designing with game theory method has received a lot of attention over last few years. Yang *et al.* [16] proposed two incentive mechanisms based on user-centric and platform-centric model. Koutsopoulos *et al.* [17] designed a mechanism considering participation levels of agents. Luo *et al.* [18] studied the incentive mechanism utilizing all-pay auction scheme. Zhao *et al.* [19] proposed two online incentive mechanisms OMZ and OMG for the scenario where agents arrived one by one. Feng *et al.* [20] proposed the auction for location-aware sensing model. Zhang *et al.* [21] designed a incentive tree mechanism to reward users for both participation and solicitation. Indeed, none of these incentive mechanisms considers the possibility of task failure. There are a few works focusing on the quality of agents or data [22]–[24], which is similar to the PoS in this paper. However, in these papers the qualities of agents or data either obey a certain probability distribution or are estimated via learning methods, which are not private information of agents. It is main difference from our model.

Zhang *et al.* [25] proposed a participant selection framework named CrowdRecruiter. It considered the probabilistic coverage constraint which is similar to our model. However, their work does not include incentive mechanism for agents. Porter *et al.* [14] investigated fault tolerant mechanism design firstly. They presented a novel VCG-like mechanism to get truthful and individually rational, which will fail for the computational intractability. Conitzer *et al.* [26] and Stein *et al.* [27] considered similar settings where execution time of agents is uncertain. Ramchurn *et al.* [28] proposed trust-based mechanisms where participants can play the role of requester and evaluate others' trust (PoS). However, they did not set the uncertainty as a private information of agents as well. Zheng *et al.* [29] proposed the mechanism aiming to achieve high probability of tasks considering single task setting and multi-task setting. However, this work only tackles with task allocation problem, which is different from our continuous task coverage problem. And the assumption that agents can only bid integral probability of success is unreasonable.

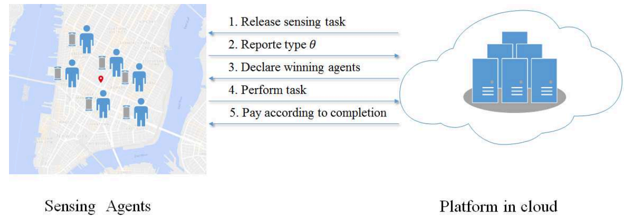

Fig. 1. A general crowdsensing system.

III. Problem Model

As illustrated by Fig. 1, a general crowdsensing system consists of two components, including a platform residing in the cloud and a set of agents with mobile devices denoted by $A = \{a_1, a_2, .., a_n\}$. The crowdsensing process can be described as follows. First, the platform releases a task T which need to be covered during a time period $[S, E]$ with a threshold probability at least T_{hr}, *e.g.*, noise monitoring on some areas from 8 am to 10 am with threshold probability 0.8. Then, there is a set A of n agents who are interested in the sensing task. Each agent $a_i \in A$ reports her private type $\theta_i = \{c_i, p_i, s_i, e_i\}$ to platform, where c_i denotes the cost of agent a_i incurred by participating in sensing task from start time s_i to end time e_i and p_i dnotes the PoS of the agent maintain. We assume that the cost c_i is declared truthfully, because we can verify it via monitoring the battery consumption, recording steps or walking routes and other methods, which is known as ex-post verification and is widely utilized in mechanism design [30]. It is noteworthy that the cost of an agent will incur no matter the agent completes sensing tasks or not, for instance, agents consume battery power to monitor noise no matter the task is completed successfully or not. Besides, we dnote p_{it} as the probability of success of agent a_i to cover the time $t \in [S, E]$. We suppose that the agent maintains the same PoS p_i during their covering periods, which indicates

$$p_{it} = \begin{cases} p_i, & t \in [s_i, e_i), \\ 0, & t \notin [s_i, e_i). \end{cases}.$$

If the task is covered by multiple agents at time t, the probability of covering task at time t is

$$p_t = 1 - \prod_{a_i \in A} (1 - p_{it}).$$

By collecting profile of agents' types, the platform selects a subset $I \subseteq A$ of agents to complete the sensing task. After completing a sensing task, an agent will receive a reward r_i. We define the utility u_i of an agent a_i as her reward r_i minus the cost c_i, *i.e.*,

$$u_i = \begin{cases} r_i - c_i, & a_i \in I, \\ 0, & \text{otherwise}. \end{cases}.$$

Due to the rational self-interest of agents, they would try to maximize their own utilities. On the contrary, the objective of platform is to maximize social welfare. The social welfare U is defined as the value of a single sensing task V minus the total cost of participants, which is presented as

$$U = V - \sum_{a_i \in I} c_i.$$

Since the value of a single task is pre-determined, the objective of the platform is to minimize total cost of selected agents, while ensuring the task are covered with a PoS at least T_{hr}, *i.e.*, we have

$$p_t \geq T_{hr}, \text{ for any } t \in [S, E] \tag{1}$$

We define the *weight* of an agent a_i as $w_i = -\log(1 - p_i)$, and let $W = -\log(1 - T_{hr})$ denote the *threshold weight*. Similarly, we let $w_{it} = -\log(1 - p_{it})$ denote the weight of an agent to cover the time t. Thus, the inequation (1) can be expressed as

$$\sum_{a_i \in I} w_{it} \geq W, \text{ for any } t \in [S, E] \tag{2}$$

As a notation convention, we use $-i$ in subscript to denote all users except i, e.g, we write types of all agents as $\theta = (\theta_i, \theta_{-i})$, and θ_{-i} denotes the types of all agents excluding agent a_i. We propose two different problem models.

A. Single Slot Coverage Model

In single slot coverage model, we divide the period $[S, T]$ into a sequence of unit slots $SL = \{sl_1, sl_2, ..., sl_k\}$. Agent $a_i \in A$ selects a slot set $SL_i \subseteq SL$, duing which she would like to participate in the sensing task. Every participant promises to cover the whole slot, if she wins the auction. Furthermore, selecting non-adjacent slots by the same agent is permitted. The result of the auction in a slot does not influence auctions in the other slots. Thus, we can focus on the auction in a single slot sl_m. The type of agent a_i reduces to $\theta_i = \{c_i, p_i\}$. Since there is only one slot, we let p_i and w_i denote p_{it} and w_{it} in this model, respectively. Therefore, we can formulate the optimization problem of selecting winners as an integer linear programing problem ILP1 using inequation (2). Let $I_m \subseteq A$ denote a set of agents who participate the auction in slot sl_m, we have:

$$\begin{aligned} \min \quad & \sum_{a_i \in I_m} c_i x_i \\ \text{s.t.} \quad & \sum_{a_i \in I_m} w_i x_i \geq W \\ & x_i \in \{0, 1\}, \qquad \forall a_i \in I_m \end{aligned} \tag{ILP1}$$

where $x_i = 1$, if agent a_i is selected, otherwise $x_i = 0$.

B. Continuous Coverage Model

In continuous coverage model, agent a_i bids type $\theta_i = \{c_i, p_i, s_i, e_i\}$. During period $[S, E]$, there are at most $2n$ different time points when agents start or stop sensing tasks. These discrete points between time point S and E are defined as *critical time points*, and we denote the set of critical time points as

$$\begin{aligned} T =& \{t | t = s_i \text{ or } t = e_i, a_i \in A, s_i \in [S, E], e_i \in [S, E]\} \\ & \cup \{S, E\}. \end{aligned}$$

If a critical time point $t \in [s_i, e_i)$, the point could be covered by the agent a_i. In other words, agent a_i is interested in covering critical time point t. Then the optimization problem reduces to the integer programming problem ILP2 as follow:

$$\begin{aligned} \min_{I \subseteq A} \quad & \sum_{a_i \in I} c_i x_i \\ \text{s.t.} \quad & \sum_{a_i \in I} w_{it} x_i \geq W, \forall t \in T \cup \{S, E\} \\ & x_i \in \{0, 1\}, \qquad \forall a_i \in I \end{aligned} \tag{ILP2}$$

where $x_i = 1$, if agent a_i is selected, otherwise $x_i = 0$.

If agent a_i is selected as a winner, she will cover the whole period $[s_i, e_i)$, *i.e.*, she wins all auctions at critical time points she could cover. We could regard these critical time points as items of an auction, and agents bid for a specified bundle of items. They prefer to win the whole bundle, and give up the auction for any other bundle. Thus, we could regard the agents as single-minded agents who are only satisfied with only winning all the critical time points during her cover period [12].

C. Economic Properties

Besides the optimization objective mentioned above, our goals include designing mechanisms to satisfy the following economical properties:

- **Computational Efficiency**: A mechanism is computational efficient if it can compute allocation and rewards in polynomial time.
- **Truthfulness (in expectation)**: A mechanism is truthful (in expectation) if agent cannot improve her (expected utility) by misreporting her type regardless of others' bids. In other words, the (expected) utility of agent a_i by reporting true type θ is larger than by reporting any other type θ'. *i.e.*, $u_i(\theta_i, \theta{-}i) \geq u_i(\theta'_i, \theta_{-i})$
- **Individual Rationality**:A mechanism is individually rational if agent a_i will get non-negative expected utility by reporting true type θ_i, *i.e.*, $u_i(\theta_i, \theta{-}i) \geq 0$

IV. Mechanism Design

Traditional pricing schemes [16]–[19] dose not consider that the success probabiliies of agents also effect the optimization solution.

To address above challenges, we design two mechanisms based on reverse auction, including winner selection algorithms and pricing schemes.

A. Single Slot Coverage Mechanism

Winner Selection: With the discussion in Section III-A, the winner selection problem can be expressed as a representive minimum knapsack problem [31] which is proved to be NP-hard. There is no polynomial algorithm to find an optimal solution yet. Instead, we design an approximation algorithm to determine winners in a greedy manner, as Algorithm 1 shows. First, agents who would participate in the auction in slot sl_m are essentially sorted in ascending according to their cost per weight (*i.e.*, $c_1/w_1 \leq c_2/w_2 \leq \ldots \leq c_n/w_n$). From Line 5 to 13, we start to check agents iteratively in the given order. If the weight requirement is not satisfied after selecting the agent a_i as a winner, we add a_i to the set of candidates I_{cur}. If the weight requirement would be satisfied after adding agent a_i to the set I_{cur}, the agent set $I_{cur} \cup \{a_i\}$ will lead to a feasible solution of ILP1. Every new feasible solution in each iteration is compared to the best solution so far and the better one is stored. With the feasible solution existing, the Algorithm 1 returns a winner set I satisfying the weight requirement.

Pricing Scheme: The reward for a winning agent is based on critical value and ex-post execution. Since the cost is reported truthfully, the critical value of an agent is defined as the minimum PoS that agent should declare to win the auction [12]. We present the details in Algorithm 2. To get the critical value of agent a_i, we temporarily set aside the winner a_i and rerun the Algorithm 1 over the rest of agents to get a new winner set I'. If agent a_i declares w_i/c_i greater than minimum w_j/c_j in I', the winner selection algorithm will select agent a_i as a winner because of the monotonicity of mechanism, of which we will provide a proof later. We can obtain the critical PoS by $\hat{p}_i = 1 - \mathrm{e}^{\hat{w}_i}$. After the winning agent tries to

Algorithm 1 SingleSlotAllocation(A, W, c, w)

Input: A set A of n users, threshold weight W, a profile of cost c, a profile of weight w
Output: A set of selected agents I
1: Order users such that $c_1/w_1 \leq c_2/w_2 \leq \ldots \leq c_i/w_i$;
2: $I \leftarrow \emptyset$;$I_{cur} \leftarrow \emptyset$;
3: $C_{Opt} \leftarrow \infty$; $C_{cur} \leftarrow 0$;
4: $W_{cur} \leftarrow 0$;
5: **for** $i \leftarrow 1$ to n **do**
6:   **if** $W_{cur} + w_i < W$ **then**
7:     $W_{cur} \leftarrow w_i + W_{cur}$; $C_{cur} \leftarrow c_i + C_{cur}$; $I_{cur} \leftarrow I_{cur} \cup \{a_i\}$;
8:   **else**
9:     **if** $C_{cur} + c_i < C_{Opt}$ **then**
10:       $C_{Opt} \leftarrow C_{cur} + c_i$; $I \leftarrow I_{cur} \cup \{a_i\}$;
11:     **end if**
12:   **end if**
13: **end for**
14: **return** I

Algorithm 2 SingleSlotReward

Input: A set A of n agents, a threshold weight W, a profile of cost c, a profile of weight w, and a user $a_i \in I$ win the auction
Output: The reward r_i of the agent a_i
$I' \leftarrow SingleSlotAllocation(A_{-i}, W, c_{-i}, w_{-i})$;
$\hat{w}_i \leftarrow \min_{a_j \in I'} c_i \frac{w_j}{c_j}$;
$\hat{p}_i \leftarrow 1 - \mathrm{e}^{-\hat{w}_i}$;
if agent complete the task **then**
  $r_i \leftarrow \beta(1 - \hat{p}_i) + c_i$;
else
  $r_i \leftarrow -\beta\hat{p}_i + c_i$;
end if
return r_i

complete the sensing task, she will receive different rewards depending on whether she complete the task successfully. If an agent a_i complete task successfully, we would pay the agent $\beta(1 - \hat{p}_i) + c_i$, otherwise she would receive $-\beta\hat{p}_i + c_i$ as a reward. The β is a pre-determine coefficient here.

We first prove that the SSC mechanism is monotone with PoS, which is indispensability for following proof. Monotonicity in PoS of mechanism implies that if user win auction with declaring PoS p_i, they will still win the auction by declaring higher PoS.

Lemma 1. *The winner selection algorithm in SSC mechanism is monotone in PoS.*

Proof: By raising PoS, the agent a_i would be inserted in a more prior order in Algorithm 1. If the agent a_i is selected by I_{cur} (Line 6), she would be selected by I_{cur} in an earlier iteration by misreporting PoS. Otherwise the set $I_{cur} \cup \{a_i\}$ was chosen as the best solution at last(Line 9). Thus, she would be added to I_{cur} or to I by reporting higher PoS. Therefore, the Algorithm 1 is monotone in PoS. ■

Lemma 2. *The SSC mechanism is computationally efficient.*

Proof: In Algorithm 1, sorting n agents takes $O(n \log n)$ time and selecting winners form agents takes $O(n)$ time. The pricing scheme runs the winner selecting algorithm once. Since there are at most n winners, the time complexity of reward calculation is $O(n^2 \log n)$. Hence, the running time of the mechanism is bounded by $O(n^2 \log n)$, implying that the SSC mechanism is computationally efficient. ■

Lemma 3. *The SSC mechanism is individually rational.*

Proof: Denote p_i as the true PoS of an agent a_i. We consider the expected utility $\bar{u}_i$ of the winner a_i. If the agent win the auction, her expected utility is $\bar{u}_i = \beta(p - \hat{p}_i)$. Obviously $\bar{u}_i$ is non-negative. And her utility is 0 when she loses the auction. To sum up, agent has a non-negative utility when she declares true type. ■

Lemma 4. *The SSC mechanism is truthful (in expectation).*

Proof: If an agent a_i wins the auction by reporting p_i, it indicates that $p_i \geq \hat{p}_i$. With discussion above, the agent, who declares a higher PoS, still wins the auction and gets the same expected utility. If the agent reduces the PoS she bids, she may probably lose the auction and get a utility of 0. Therefore, winners cannot get better expected utility by misreporting PoS.

If the agent a_i loses the auction by reporting p_i, it implies that $p_i < \hat{p}_i$. The expected utility of the agent will be negative when the agent tries to win the auction by declaring higher PoS. The utility of a losing agent is not better than the utility by biding true type.

Consequently, no matter whether the agent wins the auction or not, she has no incentive to misreport her PoS. ■

Lemma 5. *The approximation ratio of SingleSlotAllocation algorithm is 2.*

Due to the space limit, the detailed proof can be found in our online technical report [32].

B. Continuous Coverage Mechanism

Under the sing-minded setting, our mechanism CC is also based on the reverse auction, consisting of a winner selection algorithm and a pricing scheme for the continuous coverage model.

Winner Selection: The winner selection problem can be reduced from the weighted set cover probelm in polynomial time, which is already known to be NP-hard [33]. To get a near-optimal solution, we resort to the property of submodularity in the auction, which is defined as follow:

Definition 1. *(Monotone Submodular Function) Let V be a finite set. A function f: $2^V \to \mathbb{R}$ is submodular if and only if*

$$f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B),$$

for any $A \subseteq B \subseteq V$ and $v \in V \backslash B$, and f is monotonically increasing if and only if $f(A) \leq f(B)$, for any $A \subseteq B$.

Algorithm 3 ContinousCoverageAllocation(A, θ, W, S, E)

Input: A set A of n agents with type $\theta = \{c, w, s, e\}$ and a threshold weight W, the staring time S and stoping time E of the sensing task.
Output: A set of selected agents I

1: $T \leftarrow \{S, E\}$
2: **for** $i \leftarrow 1$ to n **do**
3:   **if** $s_i \in [S, E]$ **then**
4:     $T \leftarrow T \cup \{s_i\}$;
5:   **end if**
6:   **if** $e_i \in [S, E]$ **then**
7:     $T \leftarrow T \cup \{e_i\}$;
8:   **end if**
9: **end for**
10: sort T in increasing order;
11: **for all** i in T **do**
12:   $W_i \leftarrow 0$;
13: **end for**
14: **while** $\exists k : W_k < W$ **do**
15:   $a_i \leftarrow \arg\max_{a_i \in A \backslash I} \sum_{j \in T} \min(w_{ij}, W - W_j)/c_i$;
16:   $I \leftarrow I \cup \{a_i\}$;
17:   **for all** $j \in T$ **do**
18:     $W_j \leftarrow \min(W, W_j + w_{ij})$;
19:   **end for**
20: **end while**
21: **return** I;

Algorithm 4 ContinuousCoverageReward

Input: A set A of n agents with type $\theta = \{c, p, s, e\}$, a threshold weight W, a winning agent $a_i \in I$
Output: A reward r_i for a_i

$I' \leftarrow ContinousCoverageAllocation(A_{-i}, \theta_{-i}, W, S, E)$
for all $a_j \in I'$ **do**
  $\hat{w}_i \leftarrow min\{\hat{w}_i, c_i \sum_{t \in T} \min(w_{it}, W - W_k)/c_j\}$
  update all W_k
end for
$\hat{p}_i \leftarrow 1 - \mathrm{e}^{-\hat{w}_i}$
if agent complete the task **then**
  $r_i \leftarrow \beta(1 - \hat{p}_i) + c_i$;
else
  $r_i \leftarrow -\beta\hat{p}_i + c_i$;
end if
return r_i

Considering the computational intractability, we design a algorithm in a greedy manner as illustrated in Algorithm 3. First, we generate a set of critical time points according to the reporting types of candidates(Line 1 to 9). Then we let W_j denote the sum of weights caused by the agents who would cover the time point j. Agents will be selected iteratively until all W_j satisfy the threshold weight requirement. In each iteration, we select the agent with maximal ratio of total weights to cost (*i.e.*, $\sum_{j \in T} \min(w_{ij}, W - W_j)/c_i$), as Line 15 to 16 shows. Then W_j of each critical time point is updated(Line 17 to 19).

Pricing Scheme: Similar to the single slot coverage model, the reward for an agent is based on the critical value and depends the execution of agent. The critical value is also defined as the minimum PoS to ensure a agent winning. To determine the reward of a winner a_i, Algorithm 4 sets the agent a_i aside and reruns *ContinousCoverageAllocation* algorithm. However, a_i may be selected as a winner in different iterations in the winner selection algorithm. As a result, a_i may have diverse critical value in different iterations. Thus, we pick the minimal critical weight as the critical weight $\hat{w}$ (Line 2 to Line 5). Then the critical value $\hat{p_i}$ of the agent a_i can be obtained by the equation $\hat{p}_i = 1 - \mathrm{e}^{-\hat{w}_i}$. If the agent complete the sensing task over her covering period successfully, we will reward her $\beta(1-\hat{p_i}) + c_i$, otherwise she will get $-\beta\hat{p_i} + c_i$. The β is a pre-determine coefficient here.

Before discussing the properties of CC mechanism, we prove the monotonicity of the winner selection algorithm.

Lemma 6. *The winner selection algorithm in CC mechanism is monotone in PoS.*

Proof: If an agent a_i with the type $\theta_i = \{c_i, p_i, s_i, e_i\}$ is selected as a winner, a_i will still win by declaring a higher PoS. Because higher PoS helps a_i to be chosen in a earlier iteration or the same iteration accoring to Algorithm 4. ■

Lemma 7. *The CC mechanism is individually rational.*

Proof: Denote p_i as the true PoS of an agent a_i. Since all agents are single-minded, agents will cover the whole set of critical time points successfully or unsuccessfully. The expected utility of the winning agent is $\beta(p_i - \hat{p_i})$, which is non-negative when the agent declares true type. When agent lose auction, she will get a utility 0. Ultimately, the CC mechanism is individually rational. ■

Lemma 8. *The CC mechanism is truthful (in expectation).*

Proof: If agent a_i wins the auction by declaring true type, she will still win with reporting higher PoS because of the monotonicity of the winner selection algorithm. However, due to the independence between the expected utility and declaring PoS, the agent gets the same expected utility when she wins the auction. If she reports lower PoS, she takes a risk of losing auction and getting a utility 0. In conclusion, misreporting of winning agents cannot lead to a better expected utility than declaring true type.

If agent a_i loses the auction by biding truthful, she would get a negative expected utility by misreporting type to win the auction, which is not better than zero utility.

Consequently, continuous coverage mechanism is truthful. ■

Lemma 9. *The CC mechanism is computationally efficient.*

Proof: The ContinousCoverageAllocation algorithm generates the at most $2n$ critical time points, and sorts them with time bound $O(n \log n)$. The algorithm selects winners iteratively, which is executed at most $2n$ times. In each iteration, we traversal at most n agents who are interested in no more than $2n$ critical time points. Hence, the time complexity of Algorithm 3 is $O(n^3)$. Algorithm 4 runs the winner selection algorithm at most n times, whose time complexity is $O(n^4)$. Therefore, the CC mechanism is computationally efficient. ■

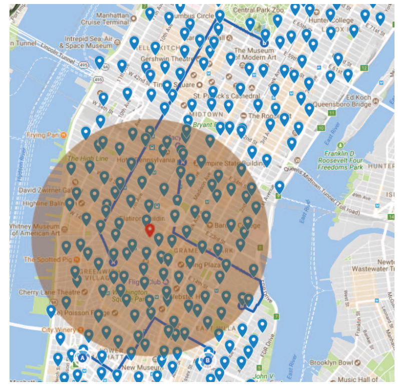

Fig. 2. Spatial distribution of the CitiBike data set.

We define the total covering weights of agents set I as a function:

$$f(I) = \sum_{j \in T} \min(\sum_{a_i \in I} w_{ij}, W),$$

which can be proved to be monotonically non-decreasing submodular. We denote $\Delta_i(S) = f(S \cup \{a_i\}) - f(S)$. We work on the assumption that there are k iterations in Algorithm 3 and let I^i denote a set of agents after the i-th iterations. Thus, I^k denotes a set of selected agents produced by Algorithm 3 terminally. Renaming the agent winning in the t-th iteration in Algorithm 3 as a_t, the coverage contribution caused by agent a_t is $c_t / \sum_{j \in T} \min(w_{tj}, W - W_j)$, which is denoted as μ^t. We first present two lemma, the proof of which is given in our online technical report [32] due to the space limit.

Lemma 10. *For any agent* $a_i \in A$ *, we have:*
$\mu^1 \Delta_i(I^0) + (\mu^2 - \mu^1)\Delta_i(I^1) + \ldots + (\mu^k - \mu^{k-1})\Delta_i(I^{k-1})$
$\leq c_i(1 + \min\{\frac{\Delta_i(I^0)}{\Delta_i(I^{k-1})}, \frac{\mu^k}{\mu^1}\})$.

Due to the interest of space, we leave the detailed proof to our technical report [21]

Lemma 11. *The approximation ratio of the ContinousCoverageAllocation algorithm is* $1 + \ln \min\left\{\max_{a_i \in A} \frac{\Delta_i(I^0)}{\Delta_i(I^{k-1})}, \frac{\mu^k}{\mu^1}\right\}$.

V. Evaluation

To evaluate the performance of our fault tolerant mechanisms closely, we implemented SSC mechanism and CC mechanism based on the trip data set of CitiBike [15] around Manhattan, New York. We present our evaluation results in this section.

A. Experimental Setup

We assume that all sharing bikes are embedded with some sensors to monitor noise, and sensors can be powered by a mini electric generator embedded on wheels. Sensors in a bicycle are regarded working during a ride and are expected to stop working when the agent returns the bike. To generate the

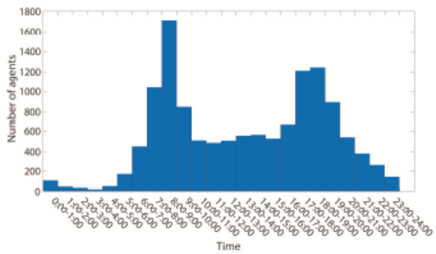

Fig. 3. Time distribution of the number of agents.

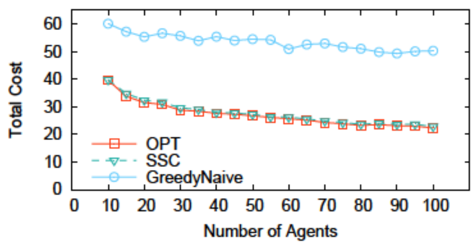

Fig. 4. Total cost in single slot coverage model.

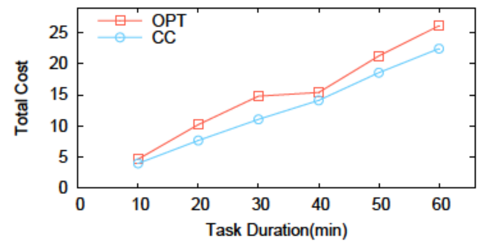

Fig. 5. Total cost in continuous coverage model.

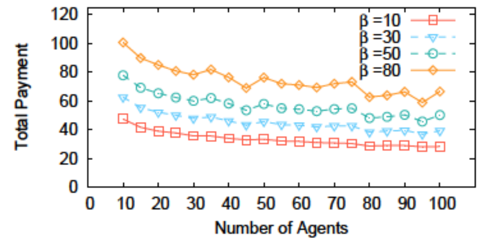

Fig. 6. Total payment in single slot coverage model.

PoS of agents, we adopt sensing model based on exponential attenuation probabilistic model in WSN [11]. We choose a certain area in Manhattan as the monitoring area, the PoS of sensors decays with the increasing distance to the monitoring area, *i.e.*, $p = e^{-\alpha d+\xi}$, where α is a decay factor influencing attenuation speed, d represents the distance between bikes and monitoring area. In our simulation, we set α fixed 0.05. Random value ξ denotes the individual difference of each agent, which is drawn from a uniform distribution on the interval $[0, 0.005]$. Since the PoS of an agent during a ride is constant, we generate the PoS of an agent by using the average distance during her ride, which is calculated according to the path produced by Google Maps Bicycle Navigation API [34]. Besides, the threshold requirement T_{hr} is fixed 0.8. We also fix the reward factor β as 10. We generate the cost of agents by multiplying riding duration by a random value sampling from the Gaussian distribution with mean 20 and variance 5.

We choose the data set in January 2017, which has more than 120,000 records. The *start time, stop time, start location (longitude and latitude)* and *stop location* of each ride are recorded by the data set. Fig. 2 illustrates the spatial distribution of the bike data, where blue points represent the bike stations which are either starting points or destinations of their rides. We denote the monitoring area as a red point. The bicycle routes are drawn with blue lines by Google Map on bicycle model. We also plot the number of users in the distribution of time in Fig. 3.

B. Impact of Agent Number and Task Duration

1) Total Cost: Firstly, we evaluate the impact of agent number on total cost in single slot coverage model. We randomly choose some agents who prefer to cover the period between 9 am and 10 am. To serve as a baseline, we design a naive heuristic algorithm named *GreedyNaive* which iteratively chooses the agent with the minimal w/c until the PoS requirement is satisfied. We also choose the optimal algorithm named *OPT* as a benchmark, which adopts the exhaustive method. The results are shown in Fig. 4. We observe that the total cost decreases slightly with the number of agents increasing. The reason is that more players in the auction lead to more high-quality agents, whose w/c is smaller.

In continuous coverage model, agents arrive and depart randomly, such that the number of agents during a certain period is uncontrollable. Therefore, we turn to the impact of task duration. We choose the whole agents who prefer to cover a period between 9 am and 10 am as candidates. We evaluate the total cost with various task durations. We choose the near-optimal solution namely *OPT* as a comparsion, which is caculated by Gurobi Optimizer [35]. Fig. 5 depicts that the total cost raises quickly while the task duration increases. We can tell that the task with longer duration needs to select more agents to ensure probability of success, leading to larger sensing cost.

2) Total Payment: In single slot coverage model, we investigate the impact of agent number in total payment with a tunable parameter β. The results are plotted in Fig. 6. We can see that the total payment decreases steadily as the number of agent increases, when β is small. With a large β, the total payment first declines dramatically, and then fluctuates. The reason is that the total cost occupies the majority of the total payment when β is small. On the contrary, the critical payment, which has positive correlation with agents number, occupies the majority of total payment with a large β.

In continuous coverage model, Fig. 7 illustrates the impact of task duration with diverse β. The payment keeps raising, while the task duration increases with all β. Besides, we further compare the total cost to complete the same sensing task in different time periods. We choose the agents who appear around 9 am, 12 am and 10 pm as candidates to complete the same sensing task, respectively. Fig. 8 shows that the total payments during all periods grow with the task duration increasing. Furthermore, the total payment around 10

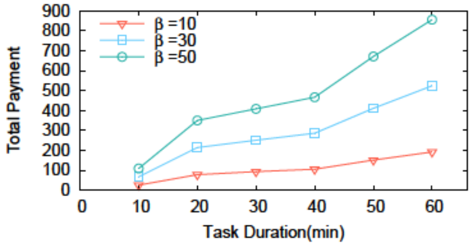

Fig. 7. Total payment with different β.

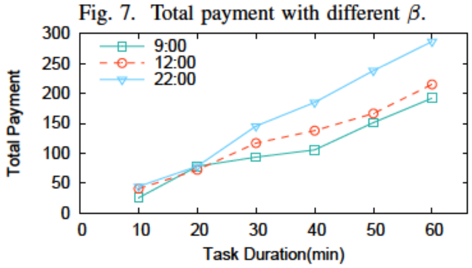

Fig. 8. Total payment during different period.

pm is always fewer than the payment around 9 am, while the payment around 12 am is the middle. In Fig. 3, we observe that most agents appear around 9 am while few agents appear around 10 pm. Therefore, we can tell that the total payment increases as the number of agents increases.

VI. Conclusion

In this paper we have made a study on incentive mechanism design of crowdsensing considering the failures of agents to complete the sensing tasks. We have presented two problem models and proposed two mechanisms with guaranteed approximation ratio and good economic properties. We have provided theoretical proof and evaluated the mechanism to show some properties of our mechanisms. As for the future works, we prefer to design a mechanism that can adaptive select new agents when the failures of tasks is detected. In addition, we would study on fault tolerant mechanism in space which guarantee the success ratio of multiple areas in space.

References

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[2] S. Kim, C. Robson, T. Zimmerman, J. Pierce, and E. M. Haber, "Creek watch: pairing usefulness and usability for successful citizen science," in *SIGCHI*, 2011.

[3] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *MobiCom*, 2012.

[4] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *MobiCom*, 2012.

[5] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones," in *SenSys*, 2009.

[6] E. Koukoumidis, L.-S. Peh, and M. R. Martonosi, "Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory," in *MobiSys*, 2011.

[7] N. Maisonneuve, M. Stevens, M. E. Niessen, and L. Steels, "Noisetube: Measuring and mapping noise pollution with mobile phones," *Information technologies in environmental engineering*, pp. 215–228, 2009.

[8] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: an end-to-end participatory urban noise mapping system," in *IPSN*, 2010.

[9] C. Wu, Z. Yang, and Y. Liu, "Smartphones based crowdsourcing for indoor localization," *IEEE Transactions on Mobile Computing*, vol. 14, no. 2, pp. 444–457, 2015.

[10] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao, "Incentives for mobile crowd sensing: A survey," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 54–67, 2016.

[11] Q. Yang, S. He, J. Li, J. Chen, and Y. Sun, "Energy-efficient probabilistic area coverage in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 367–377, 2015.

[12] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic game theory*. Cambridge University Press Cambridge, 2007.

[13] N. Nisan and A. Ronen, "Computationally feasible vcg mechanisms," *Journal of Artificial Intelligence Research*, vol. 29, pp. 19–47, 2007.

[14] R. Porter, A. Ronen, Y. Shoham, and M. Tennenholtz, "Fault tolerant mechanism design," *Artificial Intelligence*, vol. 172, no. 15, pp. 1783–1799, 2008.

[15] "Citi Bike Trip Histories," https://www.citibikenyc.com/system-data.

[16] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," 2012.

[17] I. Koutsopoulos, "Optimal incentive-driven design of participatory sensing systems," in *INFOCOM*, 2013.

[18] T. Luo, H.-P. Tan, and L. Xia, "Profit-maximizing incentive for participatory sensing," in *INFOCOM*, 2014.

[19] D. Zhao, X.-Y. Li, and H. Ma, "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint," in *INFOCOM*, 2014.

[20] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, "Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *INFOCOM*, 2014.

[21] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, "Robust incentive tree design for mobile crowdsensing," in *ICDCS*, 2017.

[22] A. Ghosh and P. McAfee, "Incentivizing high-quality user-generated content," in *WWW*, 2011.

[23] D. Peng, F. Wu, and G. Chen, "Pay as how well you do: A quality based incentive mechanism for crowdsensing," in *MobiHoc*, 2015.

[24] Z. Song, C. H. Liu, J. Wu, J. Ma, and W. Wang, "Qoi-aware multitask-oriented dynamic participant selection with budget constraints," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4618–4632, 2014.

[25] D. Zhang, H. Xiong, L. Wang, and G. Chen, "Crowdrecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *UbiComp*, 2014.

[26] V. Conitzer and A. Vidali, "Mechanism design for scheduling with uncertain execution time." in *AAAI*, 2014.

[27] S. Stein, E. H. Gerding, A. Rogers, K. Larson, and N. R. Jennings, "Algorithms and mechanisms for procuring services with uncertain durations using redundancy," *Artificial Intelligence*, vol. 175, no. 14, pp. 2021–2060, 2011.

[28] S. D. Ramchurn, C. Mezzetti, A. Giovannucci, J. A. Rodriguez-Aguilar, R. K. Dash, and N. R. Jennings, "Trust-based mechanisms for robust and efficient task allocation in the presence of execution uncertainty," *Journal of Artificial Intelligence Research*, vol. 35, no. 1, p. 119, 2009.

[29] Z. Zheng, Z. Yang, F. Wu, and G. Chen, "Mechanism design for mobile crowdsensing with execution uncertainty," in *ICDCS*, 2017.

[30] N. Nisan and A. Ronen, "Algorithmic mechanism design," *Games and Economic Behavior*, vol. 35, pp. 166–196, 2001.

[31] J. Csirik, J. B. G. Frenk, M. Labbé, and S. Zhang, "Heuristic for the 0-1 min-knapsack problem," *Acta Cybern.*, vol. 10, no. 1-2, pp. 15–20, 1991.

[32] Y. Liu, Z. Zheng, F. Wu, X. Gao, and G. Chen, "Fault tolerant mechanism design for time coverage in crowdsensing system," Tech. Rep., 2017. [Online]. Available: https://www.dropbox.com/s/x1f6r9b00v6s20a/technical%20report.pdf?dl=0

[33] M. Cygan, Ł. Kowalik, and M. Wykurz, "Exponential-time approximation of weighted set cover," *Information Processing Letters*, vol. 109, no. 16, pp. 957–961, 2009.

[34] "Google Map API," https://developers.google.com/maps/.

[35] "Gurobi Optimizer," http://www.gurobi.com/products/gurobi-optimizer.