Bi-objective Robust Incentive Mechanism Design for Mobile Crowdsensing

Jia Xu, *Member, IEEE,* Yuanhang Zhou, Yuqing Ding, Dejun Yang, *Senior Member, IEEE,* Lijie Xu

***Abstract*—In recent years, mobile crowdsensing has become an effective method for large-scale data collection. Incentive mechanism is fundamentally important for mobile crowdsensing systems. Many mobile crowdsensing systems expect to optimize multiple objectives simultaneously. Most of the existing works transform the multi-objective problem into a single objective problem through constraints or scalarization method. However, due to the uncertain importance (weights) of objectives and the instable quality of crowdsensed data, such transformation is usually unrealizable. In this paper, we aim to optimize the worst performance of two objective functions in mobile crowdsensing in order to improve the system robustness. We model an auction-based bi-objective robust mobile crowdsensing system, and design two independent objective functions to maximize the expected profit and coverage, respectively. We formulate the *Robust User Selection* (*RUS*) problem, and design an incentive mechanism, which utilizes the combination of binary search and greedy algorithm, to solve the *RUS* problem. Through both rigorous theoretical analysis and extensive simulations, we demonstrate that the designed incentive mechanisms satisfy desirable properties of computational efficiency, individual rationality, truthfulness, and constant approximation to the tightened *RUS* problem. Moreover, the proposed incentive mechanism can be easily extended to multi-objective robust mobile crowdsensing systems, and all desirable properties still hold. The simulation results reveal that our incentive mechanism achieves 11% improvement of the platform's utility, compared with the greedy algorithm for bi-objective mobile crowdsensing systems on average.**

***Keywords*—mobile crowdsensing, incentive mechanism, robustness, bi-objective problem**

I. Introduction

In recent years, as a new mode of environment sensing, data collection and information service, crowdsensing has become one of the research hotspots. With the popularization of mobile devices, such as smartphones, people can sense data of the surrounding environment through embedded sensors on smartphones in daily life. This means most smartphone users could be the participants of mobile crowdsensing. Mobile crowdsensing has the advantages of great extendibility of collecting massive data of multi-dimension for various scenarios, low requirements on users' knowledge, and low cost, etc.

Incentive mechanism design is one of the important issues in crowdsensing research. In order to stimulate more users to participate in the crowdsensing tasks, most of the existing incentive mechanisms use monetary incentives, motivating users by payment [1, 2, 3, 4]. Most of these works are auction-based incentive mechanisms [5, 6], which take into consideration the economic properties of the sensing system.

The multi-objective problem is important and pervasive in mobile crowdsensing. For example, in environmental monitoring crowdsensing, we hope that the sensing data can help us to detect any possible pollution events in time. This requires us to optimize the types of the sensors to ensure that they can provide data diversity. On the other hand, we need to optimize the locations of the mobile users so that the sensors can cover the areas as wide as possible.

In practice, many mobile crowdsensing systems want to simultaneously optimize the multiple objectives, such as position coverage for spatial phenomena observation [7], continuity of sensing data for temporal phenomena observation [8], quality of sensing data [9], and value of sensing data [10].

For the multi-objective optimization problem, the general method is transforming it to a single objective optimization problem through scalarization method, i.e., weight the objectives and then optimize the sum of weighted objectives. However, this general method is usually impractical to the mobile crowdsensing systems. First, the transformation assume that the system knows the importance of different objectives, and puts them in the unequal positions essentially. Many crowdsensing systems, such as *Ear-Phone* [11] for urban noise mapping, *Haze Watch* [12] for pollution monitoring, *SignalGuru* [13] for providing traffic information, *Frequent Trajectory Pattern Mining* [14] for activity monitoring, are developed to observe or monitor unknown events. It is difficult to determine the importance of various properties of sensing data. Second, the method of optimizing the sum of multiple weighted objectives probably makes one of the objectives very bad. This deviates our goal of system robustness. Instead of optimizing the total value of multiple objectives, we are more interested in balancing multiple functions and make sure that the worst one is as large as possible.

Another usual way to solve multi-objective optimization problem is optimizing one of the objectives and taking other objectives as constraints. However, it is hard to determine the constraint values for the single objective optimization problem. If the constraints are too loose, the problem would lose the binding on the objective. If the constraints are too tight, the performance of the solution would degrade. For most mobile crowdsensing systems, the quality of sensing data is

J. Xu, Y. Zhou, Y. Ding and L. Xu are with the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023, China. (e-mail: xujia, Q17010120, Q17010107, ljxu@njupt.edu.cn).

D. Yang. is with Colorado School of Mines, Golden, CO 80401. (e-mail: djyang@mines.edu).

This research was supported in part by NSFC grants 61872193, 61872191 and 62072254, NSF grants 1717315, and STITP grants of NJUPT SYB2019053.

instable [15, 16, 17]. This largely aggravates the uncertainty of constraint values.

Considering the uncertain importance of objectives and the instable quality of crowdsensed data, we expect that the mobile crowdsensing system can perform equally well with respect to multiple objectives. The robustness of multi-objective optimization aims to maximize the minimum value of multiple objective functions. In other words, the goal of multi-objective robustness is to optimize the worst case of multiple objective functions. This means we need to select users who are robust against the worst cases of multiple objective functions. However, there is no off-the-shelf incentive mechanism in the literature can be used for the multi-objective robust mobile crowdsensing system.

This paper aims to optimize the worst performance of two objective functions in mobile crowdsensing in order to improve the system robustness. We consider a bi-objective robust mobile crowdsensing system (can be easily extended to the multi-objective robust mobile crowdsensing system). The platform first publicizes a set of tasks in multiple geographical areas, and each task is with a value. The users can participate in mobile crowdsensing in the form of auction in different areas. The platform hopes to maximize both the expected profit of the platform and the spatial coverage of sensing data simultaneously with robustness when selecting winners. The number of winners is constrained due to the budget of the platform. The goal is to maximize the worst case of the two optimization objectives with limited number of winners. When the winners are selected, the platform notifies winners of the determination. The winners perform the tasks in their respective areas. Finally, each winner obtains the payment, which is determined by the platform.

It is very challenging to design a truthful robust incentive mechanism to maximize the minimum value of two objective functions. First, because the two objective functions are both submodular, bi-objective robustness problem is more difficult than the problem of maximizing a single submodular function, which is already NP-hard. The problem of maximizing a monotone submodular function subject to a cardinality constraint admits a $\left(1 - \frac{1}{e}\right)$ -approximation algorithm [18]. But this method cannot be used in the bi-objective robustness problem. Moreover, each user may take a strategic behavior by submitting dishonest bidding price to maximize its utility. Due to the hardness of bi-objective robustness problem, we cannot use the off-the-shelf VCG mechanism [19], which requires the optimal solution.

The main contributions of this paper are as follows:

- To the best of our knowledge, this is the first work to design bi-objective robust incentive mechanism for mobile crowdsensing, considering uncertain importance of objectives and the instable quality of crowdsensed data.
- We model an auction-based bi-objective robust mobile crowdsensing system, and design two independent objective functions to maximize the expected profit and coverage, respectively. We show that both functions of expected profit and coverage are nonnegative, monotone, and submodular.
- We formulate the *Robust User Selection* (*RUS*) problem, and design an incentive mechanism, which utilizes the combination of binary search and greedy algorithm, to solve the *RUS* problem.
- We show that the designed incentive mechanisms satisfy desirable properties of computational efficiency, individual rationality, truthfulness, and constant approximation to the tightened *RUS* problem.

The rest of the paper is organized as follows. We review the state-of-art research in Section II. Section III formulates the system models and problems, and lists some desirable properties. Section IV presents the detailed design of our incentive mechanisms. Section V presents the analysis of our incentive mechanisms. Performance evaluation is presented in Section VI. We give the discussion in Section VII. We conclude this paper in Section VIII.

II. Related Work

A. Incentive Mechanism for Mobile Crowdsensing

In location dependent mobile crowdsensing, the quality of data is largely determined by locations of mobile users. Thus the platform hopes to recruit a wide distributed participants to improve the quality of data. Jaimes *et al.* [20] designed *Maximum Coverage Algorithm* to improve the coverage of *AoIs* (*Area of Interests*) with budget constraint on the basis of [1]. Nan *et al.* [9] proposed a cross-space, multi-interaction based dynamic incentive mechanism, improving the user participation and data quality. [21, 22] used location information of users to improve the quality of sensing data while ensuring the participation. Based on [1], Zhou *et al.* [7] considered the impact of geographic locations of users on *AoI* coverage. Xu *et al.* [23] designed the incentive mechanisms for spatio-temporal tasks in mobile crowdsensing systems to minimize the social cost subject to the constraint that each of the tasks can be completed with its collective sensing time not less than a minimum sensing time required by the platform.

Although the above works used the coverage of the sensing area as a key factor of the data quality, they did not consider the difference of data importance. For example, different sensing tasks may have different values to the platform. In addition, the economic consideration of the platform is neglected. However, most mobile crowdsensing platforms do not want to incur a deficit.

The online incentive mechanisms have been studied in the literature. Zhao *et al.* proposed *OMZ* and *OMG* models, which follow the multiple-stage sampling-accepting process [24]. At every stage, the mechanism allocates tasks to a smartphone user only if its marginal density is not less than a certain density threshold that computed using previous users' information. Lin *et al.* designed Sybil-proof incentive mechanisms to deter the Sybil attack for offline crowdsensing [25] and online crowdsensing [26], respectively. Zhang *et al.* [27] considered the scenario where the mobile crowdsensing system selects workers by optimizing the completion reliability and spatial diversity of sensing tasks and designed two online incentive mechanisms based on the reverse auction. Gao *et al.* [28] presented an effective and quality-aware incentive mechanism to maximize the amount of high-quality sensing data under a limited task budget for online scenarios, where participants may arrive or leave at any random time.

However, most of the online incentive mechanisms did not consider the robustness of the crowdsensing system, that is, the level of data quality that can be achieved in the worst case.

B. Robustness in Mobile Crowdsensing

Robustness in crowdsensing indicates that sensing data quality can still be effectively guaranteed even in the worst case. Currently, there are only few studies on the robustness problem of mobile crowdsensing. [10, 29] adopted online learning to recruit participants not less than a certain number under the constraints of budget and random quality to maximize the value function. [30, 31] recruited participants with posted price, and minimized the total cost while ensuring that the expected number of participants is no less than a certain value. Qu *et al.* [32] researched on the similar scenario and made an extended discussion on the problem. Xu *et al.* [33] considered the bias between the crowdsourcers and the workers, and utilized matching technique to assign the tasks, improving the suitability of crowdsourcing. They further studied the preference over crowdsensing users, and designed truthful incentive mechanisms to minimize the social cost, such that each of the cooperative tasks can be completed by a group of compatible users [34].

Most of the existing researches on mobile crowdsensing regarded the robustness problem as the constraint of an optimization problem, or pursued the robustness of a single objective. There is no off-the-shelf research in the literature on the multi-objective robustness problem in mobile crowdsensing.

C. Multi-objective Optimization

Multi-objective optimization has been extensively studied in different fields. The traditional multi-objective optimization algorithms transform the multi-objective problem into the single-objective problem using weighted sum method, ε-constraint [35] and linear programming [36], and so on. Another technology to solve multi-objective optimization problem is evolutionary algorithms [37], such as particle swarm optimization [38]. The traditional algorithms can get one of the *Pareto optimal* [39] solutions each time, while the evolutionary algorithms can get a set of pareto optimal solutions.

However, as mentioned above, it is impractical to transform the multi-objective optimization into the constrained single objective optimization for mobile crowdsensing systems, since it is difficult to determine the importance of multi-objectives and the constraint values in mobile crowdsensing scenario.

III. System Model

We consider a mobile crowdsensing system consisting of a platform and a set U of n smartphone users, who are interested in participating in sensing tasks. The platform first publicizes a set T of t tasks in multiple geographic areas. Each task $j \in T$ is with a value v^j. Let Z be the set of areas. Each area $l \in Z$ is with a weight w_l which indicates the importance of the area. Each weight is given by the platform in advance. The platform can determine the weights based on the regional functions. For example, in environmental monitoring mobile crowdsensing, the weight of urban area is usually higher than that of suburban area, and the weight of chemical industrial area is higher than that of CBD. The users can participate in crowdsensing through auction. Let $B_i = (T_i, b_i)$ be the bid of any user $i \in U$, where $T_i \subseteq T$ is the set of tasks that i would like to perform. Note that the tasks in T_i can be distributed in multiple areas. The task set T_i can be determined based on the future schedules or daily mobility routines with little effect on user i's daily life. b_i is the bidding price of user i. Let c_i be the true cost of user i. We consider that c_i is the private information and known only to user i.

Given the task set T and the bid profile $\mathbf{B} = (B_1, B_2, \ldots, B_n)$, the platform calculates the winner set $S \subseteq U$, and notifies winners of the determination. The winners perform the sensing tasks and send data back to the platform. Each user i is paid p_i by the platform.

We define the utility of user i as the difference between the payment and its real cost:

$$u_i = \begin{cases} p_i - c_i, & if\ i \in S \\ 0, & otherwise \end{cases} \tag{1}$$

Specially, the utility of losers would be zero because they are paid nothing in our designed mechanism and there is no cost for sensing.

Note that b_i can be different from the real cost c_i because we consider the users selfish. So, the users may take a strategic behavior by claiming dishonest cost to maximize their own utilities. However, the platform only selects the profitable user i whose biding price is no more than the value it brings to platform, i.e., $b_i \le \sum_{j \in T_i} v^j, \forall i \in S$. For convenience, we assume the set U only contains these profitable users. Otherwise, we simply remove the unprofitable users from U.

The expected profit of the platform is determined by the value of tasks performed by winners and their bidding prices. We define the expected profit function $f(S)$ of platform as

$$f(S) = v(S) - \sum_{i \in S} b_i \tag{2}$$

where $v(S) = \sum_{i \in S} \sum_{j \in T_i} v^j$ is the total value of tasks performed by winners.

In addition to the expected profit, the platform also hopes the sensing data can cover as many areas as possible. We define the coverage function as

$$g(S) = \gamma \sum_{l \in Z} w_l \cdot \log(1 + n_l(S)) \tag{3}$$

where $n_l(S)$ is the number of tasks performed by the users in the set S in area l. $\gamma > 0$ is a normal coefficient to adjust the importance of area coverage as well as normalize the value of expected profit function and coverage function, so that we can pursue the robustness of system on the same magnitude. To measure the system robustness correctly, the value of normal coefficient should be set carefully. In simulations, we employ multiple random sampling to determine the value of normal coefficient. Further, the alternative parameter-free normalization method will be discussed later.

The coverage function represents the coverage level of all sensing areas. We use logarithmic function to describe the diminishing return of platform's revenue with the increase of number of users in the same area.

We define the utility of the platform u_0 as the minimum of the expected profit function and coverage function.

$$u_0 = \min\left\{f\left(S\right), g\left(S\right)\right\} \tag{4}$$

The incentive mechanism $\mathcal{M}\left(T, \mathbf{B}\right)$ outputs a winner set S and a payment profile $\mathbf{p} = \left(p_1, p_2, \ldots, p_n\right)$. The objective is maximizing the utility of platform subject to the constraint that the number of winners is no more than m, which depends on the budget of the platform. We call this problem as *RUS* problem, which can be formulated as follows:

$$\max\nolimits_{S\subseteq U} \min\left\{f\left(S\right), g\left(S\right)\right\}, \quad s.t. \quad |S| \le m \tag{P1}$$

Different from the methods reviewed in section II-D, we optimize the bi-objective problem through maximizing the minimum value of expected profit function and coverage function. It is reasonable in mobile crowdsensing systems since our goal is to achieve the robustness.

Note that the coverage function defined in (3) is a normalized coverage function. To maximize the utility of platform, i.e., the robustness of the system, γ should be set to make the values of expected profit function and coverage function as equal as possible. This is because if we take the worst of two equally important objectives (after normalized) as the robustness of the system, the values of two objectives should be as equal as possible. In our simulations, we determine the value of normal coefficient through multiple random sampling, which will be presented in section VI-A. The other possible normalization method will be discussed further in section VII.

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

- **Computational Efficiency:** An incentive mechanism is computationally efficient if the winner set S and the payment $\mathbf{p}$ can be computed in polynomial time.
- **Individual Rationality:** Each user will have a nonnegative utility while reporting true private information, i.e., $u_i \ge 0, \ \forall i \in U$.
- **Truthfulness:** A mechanism is truthful if no user can improve its utility by submitting false cost, no matter what others submit. In other words, reporting the real cost is a weakly dominant strategy for all users.
- **Constant Approximation:** The goal of the mechanism is to maximize the utility of platform. If $k \ge \theta k^*$, where k is the worst output of incentive mechanism for P1, k^* is the optimal solution of P1', we say that the incentive mechanism is θ - approximation to P1'. Specifically, if θ is a constant, we say the incentive mechanism is constant approximation to P1'.

The importance of the first two properties is obvious, because they together ensure the feasibility of an incentive mechanism. The last two properties are indispensable for guaranteeing the compatibility and high performance. Being truthful, the incentive mechanism can eliminate the fear of market manipulation and the overhead of strategizing over others for the participating users.

We list the frequently used notations in Table. I.

TABLE I: Frequently Used Notations

Symbol	Description
U, S, n	set of users, set of winners, number of users
T, T_i, t	set of tasks, set of user i's tasks, number of tasks
m	maximum number of winners
$v^j, v(S)$	value of task j, total value of winners
b_i, c_i	bidding price of user i, cost of user i
$\mathbf{B}, B_i$	bid profile, bid of user i
$\mathbf{p}, p_i$	payment profile, payment of user i
f, g	expected profit function, coverage function
f_{norm}	normalized expected profit function
g_{norm}	normalized coverage function
u_i, u_0	utility of user i, utility of platform
γ, α, ϵ	normal coefficient, relaxation coefficient, search accuracy
Z	set of areas
w_j	weight of area j
$n_j(S)$	number of winners in area j

IV. Incentive Mechanism Design

In this section, we present the *incentive Mechanism for Robust User Selection* (*MRUS*) to solve the *RUS* problem defined in (P1).

First, we give the following definition.

Definition 1. *(Nonnegative, monotone, and submodular function): Given a finite ground set V, a real-valued set function defined as* $F : 2^V \to \mathbb{R}$ *,F is called nonnegative, monotone, and submodular if and only if it satisfies the following conditions, respectively:*

- $F\left(\varnothing\right) = 0$ and $F\left(A\right) \ge 0$ for all $A \subseteq V$;
- $F\left(A\right) \le F\left(B\right)$ for all $A \subseteq B \subseteq V$;
- $F\left(A\right) + F\left(B\right) \ge F\left(A \cup B\right) + F\left(A \cap B\right)$ for any $A, B \subseteq V$ or: $F\left(A \cup \{e\}\right) - F\left(A\right) \ge F\left(B \cup \{e\}\right) - F\left(B\right)$ for all $A \subseteq B \subseteq V$ and $e \in V \backslash B$.

According to the definition of submodular function, we have the following conclusions.

Theorem 1. *The expected profit function f is a nonnegative, monotone, and submodular function.*

Proof: Since $b_i \le \sum\limits_{j\in T_i} v^j, \forall i \in S$, we have $f\left(S\right) = v\left(S\right) - \sum\limits_{i\in S} b_i = \sum\limits_{i\in S}\left(\sum\limits_{j\in T_i} v^j - b_i\right) \ge 0$. Thus, f is nonnegative.

For all $A \subseteq B \subseteq V$, we have

$$\begin{gathered} f\left(B\right) - f\left(A\right) \\ = \left(v\left(B\right) - \sum_{i\in B} b_i\right) - \left(v\left(A\right) - \sum_{i\in A} b_i\right) \\ = v\left(B\backslash A\right) - \sum_{i\in B\backslash A} b_i \\ = \sum_{i\in B\backslash A}\left(\sum_{j\in T_i} v^j - b_i\right) \ge 0 \end{gathered}$$

Thus, f is monotone.

For all $A \subseteq B \subseteq V$ and $e \in V \backslash B$, we have

$$\begin{gathered} f\left(A \cup \{e\}\right) - f\left(A\right) \\ = \left(v\left(A \cup \{e\}\right) - \sum_{i\in A\cup\{e\}} b_i\right) - \left(v\left(A\right) - \sum_{i\in A} b_i\right) \end{gathered}$$

$$= v(\{e\}) - b_e = f(B \cup \{e\}) - f(B)$$

Thus, f is submodular. ■

Theorem 2. *The coverage function g is a nonnegative, monotone, and submodular function.*

Proof: Based on equation (3), the nonnegativity of g is obvious. The monotonicity of g is also obvious as adding a new user into S cannot decrease the value of g.

For all $A \subseteq B \subseteq V$ and $e \in V \backslash B$, we have

$$g(A \cup \{e\}) - g(A)$$
$$= \gamma \sum_{l \in Z} w_l \cdot \log(1 + n_l(A \cup \{e\})) \; - \gamma \sum_{l \in Z} w_l \cdot \log(1 + n_l(A))$$
$$= \gamma \sum_{l \in Z} w_l \cdot \log\left(\frac{1+n_l(A \cup \{e\})}{1+n_l(A)}\right)$$

Let $T^l(A)$ be the task set in area l performed by users in S. Given set A, define $T_e^l(A)$ as the set of new tasks in area l performed by user e, i.e., $T_e^l(A) = T^l(A \cup \{e\}) - T^l(A)$.

There are two cases for each area $l \in Z$:

(1) $T_e^l(A) \neq \varnothing$. We have

$$n_l(A \cup \{e\}) = n_l(A) + \left|T_e^l(A)\right|,$$
$$n_l(B \cup \{e\}) = n_l(B) + \left|T_e^l(B)\right|.$$

So

$$\log\left(\frac{1+n_l(A \cup \{e\})}{1+n_l(A)}\right) = \log\left(\frac{1+n_l(A)+\left|T_e^l(A)\right|}{1+n_l(A)}\right) = \log\left(1 + \frac{\left|T_e^l(A)\right|}{1+n_l(A)}\right).$$

Since $A \subseteq B \subseteq V$, we have

$$n_l(A) \leq n_l(B) \text{ and } \left|T_e^l(A)\right| \geq \left|T_e^l(B)\right|$$

for any area $l \in Z$. Thus

$$\log\left(1 + \frac{\left|T_e^l(A)\right|}{1+n_l(A)}\right) \geq \log\left(1 + \frac{\left|T_e^l(B)\right|}{1+n_l(B)}\right) = \log\left(\frac{1+n_l(B)+\left|T_e^l(B)\right|}{1+n_l(B)}\right).$$

(2) $T_e^l(A) = \varnothing$. We have

$n_l(A \cup \{e\}) = n_l(A)$ and $n_l(B \cup \{e\}) = n_l(B)$. So

$$\log\left(\frac{1+n_l(A \cup \{e\})}{1+n_l(A)}\right) = \log\left(\frac{1+n_l(B \cup \{e\})}{1+n_l(B)}\right) = 0.$$

As a conclusion, we have

$$g(A \cup \{e\}) - g(A)$$
$$= \gamma \sum_{l \in Z} w_l \cdot \log\left(\frac{1+n_l(A \cup \{e\})}{1+n_l(A)}\right)$$
$$\geq \gamma \sum_{l \in Z} w_l \cdot \log\left(\frac{1+n_l(B \cup \{e\})}{1+n_l(B)}\right)$$
$$= g(B \cup \{e\}) - g(B)$$

Thus, g is submodular. ■

Since maximizing a submodular function is NP-hard [18], the *RUS* problem, that is, maximizing the minimum of two submodular functions, is also NP-hard.

The greedy algorithm performs well for the single-objective optimization problem, while it has bad performance for the *RUS* problem. We show that the greedy algorithm works arbitrarily badly for the example illustrated in Fig.1. Consider two additive functions (the special case of submodular function) $F_1(S) = \sum_{x \in S} F_1(\{x\})$ and $F_2(S) = \sum_{x \in S} F_2(\{x\})$, where $F_1(\{x\}) = 1/x$ and $F_2(\{x\}) = x$. The objective is to select a subset S of ground set $U = \{a, b, c, d\}$ to maximize $\min\{F_1(S), F_2(S)\}$ subject to $|S| \leq 2$. Let $1 < c < \frac{1}{b} < \frac{1}{a}$, $d \to 0$, $a \to 0$, $d \to \infty$. The values of $F_1(S), F_2(S)$, and $\min\{F_1(S), F_2(S)\}$ for all possible S are given in Table II.

Obviously, the greedy algorithm will first select c since $1/c$ is the largest value of objective function. Then the greedy algorithm will select the second one from a, b or d, and the value of objective function will be $a+c$ or $\min\{1/b + 1/c, b + c\}$ or $1/c+1/d$. However, the optimal solution is $S = \{a, d\}$, and the value of $\min\{F_1(S), F_2(S)\}$ is $\min\{1/a + 1/d, a + d\}$, which tends to be infinite. In this case, the approximation ratio of greedy algorithm tends to zero.

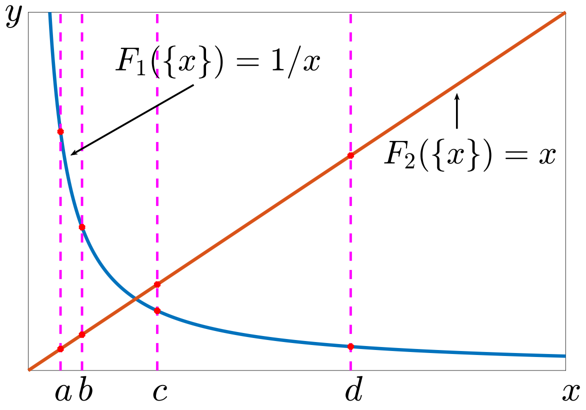


Fig. 1: Example shows that the greedy algorithm works arbitrarily badly.

TABLE II: Results of the Example Given in Fig.1

S	$F_1(S)$	$F_2(S)$	$\min(F_1(S), F_2(S))$
$\varnothing$	0	0	0
$\{a\}$	$1/a$	a	a
$\{b\}$	$1/b$	b	b
$\{c\}$	$1/c$	c	$1/c$
$\{d\}$	$1/d$	d	$1/d$
$\{a, b\}$	$1/a + 1/b$	$a + b$	$a + b$
$\{a, c\}$	$1/a + 1/a$	$a + c$	$a + c$
$\{a, d\}$	$1/a + 1/d$	$a + d$	$min((1/a + 1/d), a + d)$
$\{b, c\}$	$1/b + 1/c$	$b + c$	$min((1/b + 1/c), b + c)$
$\{b, d\}$	$1/b + 1/d$	$b + d$	$1/b + 1/d$
$\{c, d\}$	$1/c + 1/d$	$c + d$	$1/c + 1/d$

The failure of guaranteeing approximation makes the greedy algorithm less attractive. We redefine the *RUS* problem as follows:

$$\begin{aligned} &\max k \\ s.t. \quad & f(S) \geq k, g(S) \geq k, |S| \leq m, S \subseteq U \end{aligned} \tag{P2}$$

In P2, the objective is to find a set S, which maximizes the lower bounds of both $f(S)$ and $g(S)$, with the maximum size m. Obviously, P1 and P2 are equivalent.

In view of the hardness of solving P1 or P2 directly, we break the constraint of set size of P2. We formulate the relaxed *RUS* problem as

$$\begin{aligned} &\max k \\ s.t. \quad & f(S) \geq k, g(S) \geq k, |S| \leq \alpha m, S \subseteq U \end{aligned} \tag{P3}$$

where $\alpha \geq 1$ is the relaxation coefficient. In particular, when $\alpha = 1$, the relaxed *RUS* problem is equivalent to the original *RUS* problem.

In order to solve P3, for any given value of k, we find the smallest set S_k, that is,

$$S_k = \arg\min_{S \subseteq U} |S|, \quad s.t. \;\; f(S) \geq k, g(S) \geq k \tag{P4}$$

In P4, we find a smallest set S_k, which satisfies $f(S_k) \geq k, g(S_k) \geq k$, for any fixed k. If the size of S_k is not larger than αm, then the given k satisfies the constraints in P3. In other words, k is a feasible solution of P3. We call P4 as k-test problem of P3.

For any given value of k, if there is optimal algorithm or approximation algorithm of P4, we can use binary search to find the maximum value of k under the constraints for given accuracy level. In each round of binary search, for a given k, we solve P4 to find the best S_k, and check whether this k can satisfy the constraints in P3, so as to determine the direction of the next binary search round. Thus, we turn our attention to P4.

To solve P4, we define the following functions:

$$\widehat{F_{f,k}}(S) = \min\{f(S), k\} \tag{5}$$

$$\widehat{F_{g,k}}(S) = \min\{g(S), k\} \tag{6}$$

$$\overline{F_k}(S) = \frac{1}{2}\left(\widehat{F_{f,k}}(S) + \widehat{F_{g,k}}(S)\right) \tag{7}$$

Then, we have the following result.

Theorem 3. *Given any fixed k, function $\overline{F_k}$ is monotone and submodular.*

Proof: Based on Theorem 1, we have $f(B) \geq f(A)$ for all $A \subseteq B \subseteq V$. We consider the following three cases:

(1) $f(B) \geq f(A) \geq k$.

We have $\widehat{F_{f,k}}(B) = \widehat{F_{f,k}}(A) = k$.

(2) $k \geq f(B) \geq f(A)$.

We have $\widehat{F_{f,k}}(B) = f(B)$, $\widehat{F_{f,k}}(A) = f(A)$.

(3) $f(B) \geq k \geq f(A)$.

We have $\widehat{F_{f,k}}(B) = k$, $\widehat{F_{f,k}}(A) = f(A)$.

Hence, we have $\widehat{F_{f,k}}(B) \geq \widehat{F_{f,k}}(A)$ for all three cases, and conclude that function $\widehat{F_{f,k}}$ is monotone.

Next, we show $\widehat{F_{f,k}}$ is submodular. It suffices to prove that

$$\widehat{F_{f,k}}(A \cup \{e\}) - \widehat{F_{f,k}}(A) \geq \widehat{F_{f,k}}(B \cup \{e\}) - \widehat{F_{f,k}}(B) \tag{8}$$

for all $A \subseteq B \subseteq V$ and $e \in V \backslash B$.

We consider the following three cases:

(1) $\widehat{F_{f,k}}(A) = k$.

We have $\widehat{F_{f,k}}(A \cup \{e\}) = \widehat{F_{f,k}}(B) = \widehat{F_{f,k}}(B \cup \{e\}) = k$ because of the monotonicity of $\widehat{F_{f,k}}$. Inequality (8) holds.

(2) $\widehat{F_{f,k}}(A) = \hat{u}(A)$, $\widehat{F_{f,k}}(A \cup \{e\}) = k$.

We have $\widehat{F_{f,k}}(B \cup \{e\}) = k$ and $\widehat{F_{f,k}}(B) \geq \widehat{F_{f,k}}(A)$ because of the monotonicity of $\widehat{F_{f,k}}$. Inequality (8) holds.

(3) $\widehat{F_{f,k}}(A) = f(A)$, $\widehat{F_{f,k}}(A \cup \{e\}) = f(A \cup \{e\})$.

Then $\widehat{F_{f,k}}(B \cup \{e\}) - \widehat{F_{f,k}}(B)$ reaches the maximum when $\widehat{F_{f,k}}(B \cup \{e\}) = f(B \cup \{e\})$ and $\widehat{F_{f,k}}(B) = f(B)$. Since function f is submodular, we have $f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)$. Inequality (8) holds.

To sum up, $\widehat{F_{f,k}}$ is submodular.

Similarly, we can obtain that $\widehat{F_{g,k}}$ is submodular.

Thus $\overline{F_k}$, the linear function of $\widehat{F_{f,k}}$ and $\widehat{F_{g,k}}$, is monotone and submodular. ■

Obviously, $\overline{F_k}(S) = k$ if and only if $f(S) \geq k, g(S) \geq k$. This means that the test problem of k given in P4 can be redefined to finding the smallest user set S_k satisfying $\overline{F_k}(S) = k$. From the monotonicity of $\overline{F_k}(S)$ and the value range of k,we have $k = \overline{F_k}(U)$. Then, we can reformulate P4 as:

$$S_k = \arg\min_{S \subseteq U} |S|, \qquad s.t. \ \ \overline{F_k}(S) = \overline{F_k}(U) \tag{P5}$$

P5 is an instance of such a *submodular covering problem* [40], which is also NP-hard.

Fortunately, Wolsey showed that the greedy algorithm can output the solution with guaranteed approximation for P5 [41].

Theorem 4. *Given a monotonic submodular function F on a ground set V, the greedy algorithm that applied to the optimization problem:*

$$\min_{S \subseteq U} |S| \ \ such\ that\ F(S) = F(V)$$

can approximate the optimal solution within a factor of $1 + \log\left(\max_{e \in V} F(\{e\})\right)$.

Theorem 4 means that the greedy algorithm can output the winner set S with size of no more than $m(1 + \log\left(\max_{e \in U} \overline{F_k}(\{e\})\right))$.

In order to use the greedy algorithm in the inner loop of binary search over k, we need to make sure that the approximation guarantee for greedy algorithm is independent of k. This can be achieved by choosing a larger approximation guarantee. We set

$$\begin{aligned} \alpha &= 1 + \log\left(\max_{e \in U}\left(f(\{e\}) + g(\{e\})\right)\right) \\ &\geq 1 + \log\left(2\max_{e \in U} \overline{F_k}(\{e\})\right) \\ &> 1 + \log\left(\max_{e \in U} \overline{F_k}(\{e\})\right) \end{aligned}$$

So far, we have found an approximation algorithm to solve the relaxed version (P3) of the original *RUS* problem (P1) with the approximation ratio of relaxation coefficient α. We can transform P3 back into P2 by tightening the constraint from $|S| \leq \alpha m$ to $|S| \leq m$, and then use the greedy algorithm to solve P2. Fig.2. summarizes the whole problem transformations.

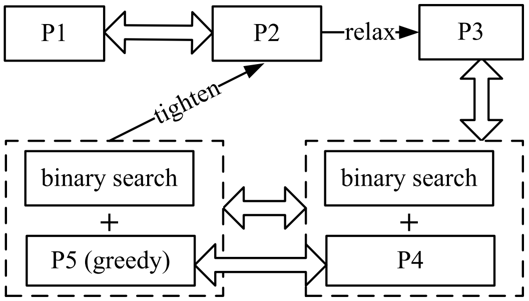


Fig. 2: Problem transforms

Now, we present the details of our *incentive Mechanism for Robust User Selection* (*MRUS*). As illustrated in Algorithm 1, *MRUS* consists of winner selection phase and payment determination phase.

Based on the definition of *RUS* problem and the monotonicity of $f(S)$ and $g(S)$, we can set the initial value of k in binary search as $k_{min} = 0$ and $k_{max} = \min\{f(U), g(U)\}$, where k_{min} is the lower bound and k_{max} is the upper bound.

In the winner selection phase, the outer while-loop (Lines 2-11) is a process of finding the maximum of k through binary search such as the number of winners is not more than m. Let S' be the winner set selected in each round of search. The inner while-loop (Lines 4-7) greedily select the users from U to S' until $\overline{F_k}(S') \geq k$. In each iteration, we select the user with maximum marginal contribution of function $\overline{F_k}(S')$ over the unselected user set $U \backslash S'$ as the winner (Lines 5-6). The binary search terminates when $(k_{max} - k_{min}) < \epsilon$, where $\epsilon \in (0, 1)$ is the search accuracy.

Algorithm 1 *MRUS*

```
Input: $f$, $g$, $m$, $\epsilon$, $U$, $T$, $\mathbf{B}$
    //winner selection
 1: $k_{min} \leftarrow 0$;  $k_{max} \leftarrow min\left(f\left(U\right), g\left(U\right)\right)$;  $S \leftarrow \varnothing$;
 2: while $(k_{max} - k_{min}) \geq \epsilon$ do
 3:   $k \leftarrow (k_{min} + k_{max})/2$; $S' \leftarrow \varnothing$;
 4:   while $\overline{F_k}(S') < k$ and $S' \neq U$ do
 5:     $i \leftarrow \arg \max_{e \in U \backslash S'} \left(\overline{F_k}(S' \cup \{e\}) - \overline{F_k}(S')\right)$;
 6:     $S' \leftarrow S' \cup \{i\}$;
 7:   end while
 8:   if $|S'| > m$ then $k_{max} \leftarrow k$;
 9:   else $k_{min} \leftarrow k$; $S \leftarrow S'$;
10:   end if
11: end while
12: $k \leftarrow k_{min}$;
    //payment determination
13: foreach $i \in U$ do $p_i \leftarrow 0$;
14: foreach $i \in S$ do
15:   $U' \leftarrow U \backslash \{i\}$;  $S' \leftarrow \varnothing$;
16:   while $\overline{F_k}(S') < k$ do
17:     $i_e \leftarrow \arg \max_{e \in U' \backslash S'} \left(\overline{F_k}(S' \cup \{e\}) - \overline{F_k}(S')\right)$;
18:     if $f(S' \cup \{i\}) < k$ and $f(S' \cup \{i_e\}) < k$ then
19:       $p'_{i_e} \leftarrow v(\{i\}) - v(\{i_e\}) + \widehat{F_{g,k}}(S' \cup \{i\})$
          $- \widehat{F_{g,k}}(S' \cup \{i_e\}) + b_{i_e}$;
20:     if $f(S' \cup \{i\}) < k$ and $f(S' \cup \{i_e\}) \geq k$ then
21:       $p'_{i_e} \leftarrow f(S') + v(\{i\}) + \widehat{F_{g,k}}(S' \cup \{i\})$
          $- \widehat{F_{g,k}}(S' \cup \{i_e\}) - k$;
22:     if $f(S' \cup \{i\}) \geq k$ and $f(S' \cup \{i_e\}) \geq k$ then
23:       $p'_{i_e} \leftarrow \max\{f(S') + v(\{i\}) - k,\ f(S') + v(\{i\})$
          $+ \widehat{F_{g,k}}(S' \cup \{i\}) - \widehat{F_{g,k}}(S' \cup \{i_e\}) - k\}$;
24:     if $f(S' \cup \{i\}) \geq k$ and $f(S' \cup \{i_e\}) < k$ then
25:       $p'_{i_e} \leftarrow \max\{f(S') + v(\{i\}) - k,\ v(\{i\}) - v(\{i_e\})$
          $+ \widehat{F_{g,k}}(S' \cup \{i\}) - \widehat{F_{g,k}}(S' \cup \{i_e\}) + b_{i_e}\}$;
26:     end if
27:     $p_i \leftarrow \max\{p_i, p'_{i_e}\}$;
28:     $S' \leftarrow S' \cup \{i_e\}$;
29:   end while
30: end for
31: return $(S, \mathbf{p})$;
```

In payment determination phase, for each winner $i \in S$, we execute the winner selection phase over $U \backslash \{i\}$, and the winner set is denoted by S'. We compute the maximum price that user i can be selected instead of each user in S'. Specifically, we consider four cases. In each case, let p'_{i_e} be the critical payment to user i for any replacement i_e. Finally, we set $p_i = \min\left\{\max_{i_e \in U' \backslash S'} p'_{i_e}, v(\{i\})\right\}$. We will prove that this price is a critical payment for user i later.

V. Mechanism Analysis

In the following, we present theoretical analysis, demonstrating that *MRUS* can achieve the desirable properties of computational efficiency, individual rationality, truthfulness and constant approximation.

Lemma 1. *MRUS is computationally efficient.*

Proof: Based on line 1 of *MRUS*, initially, $k_{max} - k_{min} = min\left(f\left(U\right), g\left(U\right)\right)$. Considering the search accuracy ϵ, the binary search (Lines 2-11) has $log\frac{min(f(U),g(U))}{\epsilon}$ iterations. In each iteration, finding the user with maximum marginal contribution (Line 5) takes $O(n)$ time. Since there are at most n users, the while-loop (Lines 4-7) takes $O\left(n^2\right)$ time. Thus, the winner selection phase takes $O\left(n^2 log\frac{min(f(U),g(U))}{\epsilon}\right)$ time. In each iteration of the for-loop (Lines 13–29), a process similar to line 4–7 is executed. Hence the running time of payment determination phase is $O\left(n^3\right)$. Hence the running time of *MRUS* is $O\left(\max\left\{n^2 log\frac{\min(f(U),g(U))}{\epsilon}, n^3\right\}\right)$. ■

Lemma 2. *MRUS is individually rational.*

Proof: Let i_e be user i's replacement which appears in the ith place in the sorting over $U \backslash \{i\}$. Since user i_e would not be at the i th place if i is considered, we have $\overline{F_k}(S \cup \{i\}) - \overline{F_k}(S) \geq \overline{F_k}(S \cup \{i_e\}) - \overline{F_k}(S)$, i.e., $\overline{F_k}(S \cup \{i\}) \geq \overline{F_k}(S \cup \{i_e\})$. Based on inequation (8), we have:

$$\begin{aligned} &\widehat{F_{f,k}}(S \cup \{i\}) + \widehat{F_{g,k}}(S \cup \{i\}) \\ &\geq \widehat{F_{f,k}}(S \cup \{i_e\}) + \widehat{F_{g,k}}(S \cup \{i_e\}) \end{aligned} \tag{9}$$

We consider the following four cases as the payment determination phase considers:

(1) $f(S \cup \{i\}) < k$, $f(S \cup \{i_e\}) < k$.

Substitute the conditions into inequation (9), we have

$$\begin{aligned} b_i &\leq v(\{i\}) - v(\{i_e\}) + \widehat{F_{g,k}}(S \cup \{i\}) - \widehat{F_{g,k}}(S \cup \{i_e\}) + b_{i_e} \\ &= v(\{i\}) - v(\{i_e\}) + \widehat{F_{g,k}}(S' \cup \{i\}) - \widehat{F_{g,k}}(S' \cup \{i_e\}) + b_{i_e} \\ &= p'_{i_e} \end{aligned}$$

where the first equality relies on the observation that $S = S'$ for every $e \leq i$ in the payment determination phase.

(2) $f(S \cup \{i\}) < k$, $f(S \cup \{i_e\}) \geq k$.

Substitute the conditions into inequation (9), we have

$$\begin{aligned} b_i &\leq f(S) + v(\{i\}) + \widehat{F_{g,k}}(S \cup \{i\}) - \widehat{F_{g,k}}(S \cup \{i_e\}) - k \\ &= f(S') + v(\{i\}) + \widehat{F_{g,k}}(S' \cup \{i\}) - \widehat{F_{g,k}}(S' \cup \{i_e\}) - k \\ &= p'_{i_e} \end{aligned}$$

(3) $f(S \cup \{i\}) \geq k$, $f(S \cup \{i_e\}) \geq k$.

Substitute the conditions into inequation (9), we have

$$\begin{aligned} b_i &\leq \max\{f(S) + v(\{i\}) - k, f(S) + v(\{i\}) \\ &\quad + \widehat{F_{g,k}}(S \cup \{i\}) - \widehat{F_{g,k}}(S \cup \{i_e\}) - k\} \\ &= \max\{f(S') + v(\{i\}) - k, f(S') + v(\{i\}) \\ &\quad + \widehat{F_{g,k}}(S' \cup \{i\}) - \widehat{F_{g,k}}(S' \cup \{i_e\}) - k\} \\ &= p'_{i_e} \end{aligned}$$

(4) $f(S \cup \{i\}) \geq k$, $f(S \cup \{i_e\}) < k$.

Substitute the conditions into inequation (9), we have

$$\begin{aligned} b_i &\leq \max\{f(S) + v(\{i\}) - k, v(\{i\}) - v(\{i_e\}) \\ &\quad + \widehat{F_{g,k}}(S \cup \{i\}) - \widehat{F_{g,k}}(S \cup \{i_e\}) + b_{i_e}\} \\ &= \max\{f(S') + v(\{i\}) - k, v(\{i\}) - v(\{i_e\}) \end{aligned}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3072953, IEEE Internet of Things Journal

$$\begin{aligned}&+\widehat{F_{g,k}}\left(S'\cup\{i\}\right)-\widehat{F_{g,k}}\left(S'\cup\{i_e\}\right)+b_{i_e}\}\\&=p'_{i_e}\end{aligned}$$

Since $p_i=\max_{i_e\in U'\setminus S'}p'_{i_e}$, we have $b_i\le p_i$. ■

Before analyzing the truthfulness of *MRUS*, we first introduce the Myerson's Theorem [42].

Theorem 5 [8, Theorem 2]: *An auction mechanism is truthful if and only if:*

- *The selection rule is monotone: If user i wins the auction by bidding b_i, it also wins by bidding $b_i^{'}\le b_i$;*
- *Each winner is paid the critical value: User i would not win the auction if it bids higher than this value.*

Lemma 3. *MRUS is truthful.*

Proof: Based on Theorem 5, it suffices to prove that the selection rule of *MRUS* is monotone and the payment p_i for each i is the critical value. The monotonicity of selection rule is obvious as bidding a smaller value cannot push user i backwards in the sorting. We next show that p_i is the critical value for i in the sense that bidding higher p_i could prevent i from winning the auction. Note that $p_i=\max_{e\in\{1,\ldots,L\}}p'_{i_e}$, where L is the number of winners in the payment determination phase. Again, we consider the following four cases as the payment determination phase considers:

(1) $f\left(S'\cup\{i\}\right)<k,\ f\left(S'\cup\{i_e\}\right)<k$.

In this case, let $b_i>p'_{i_e}$, i.e.,

$$\begin{gathered}b_i>v\left(\{i\}\right)-v\left(\{i_e\}\right)+\widehat{F_{g,k}}\left(S'\cup\{i\}\right)-\widehat{F_{g,k}}\left(S'\cup\{i_e\}\right)+b_{i_e}\\ \Rightarrow v\left(\{i\}\right)-b_i+\widehat{F_{g,k}}\left(S'\cup\{i\}\right)<v\left(\{i_e\}\right)-b_{i_e}+\\ \widehat{F_{g,k}}\left(S'\cup\{i_e\}\right)\\ \Rightarrow v\left(S'\cup\{i\}\right)-\sum_{i'\in S'\cup\{i\}}b_{i'}+\widehat{F_{g,k}}\left(S'\cup\{i\}\right)<\\ v\left(S'\cup\{i_e\}\right)-\sum_{i'\in S'\cup\{i_e\}}b_{i'}+\widehat{F_{g,k}}\left(S'\cup\{i_e\}\right)\\ \Rightarrow f\left(S'\cup\{i\}\right)+\widehat{F_{g,k}}\left(S'\cup\{i\}\right)<f\left(S'\cup\{i_e\}\right)+\\ \widehat{F_{g,k}}\left(S'\cup\{i_e\}\right)\\ \Rightarrow\widehat{F_{f,k}}\left(S'\cup\{i\}\right)+\widehat{F_{g,k}}\left(S'\cup\{i\}\right)<\widehat{F_{f,k}}\left(S'\cup\{i_e\}\right)+\\ \widehat{F_{g,k}}\left(S'\cup\{i_e\}\right)\\ \Rightarrow\overline{F_k}\left(S'\cup\{i\}\right)<\overline{F_k}\left(S'\cup\{i_e\}\right)\\ \Rightarrow\overline{F_k}\left(S'\cup\{i\}\right)-\overline{F_k}\left(S'\right)<\overline{F_k}\left(S'\cup\{i_e\}\right)-\overline{F_k}\left(S'\right)\end{gathered}$$

This means user i will be replaced by i_e according to the selection rule of *MRUS*.

The same result can be obtained in other three cases. We give the proof as follows:

(2) $\left(S'\cup\{i\}\right)<k,\ f\left(S'\cup\{i_e\}\right)\ge k$.

In this case, Let $b_i>p'_{i_e}$, i.e.,

$$\begin{gathered}b_i>f\left(S'\right)+v\left(\{i\}\right)+\widehat{F_{g,k}}\left(S'\cup\{i\}\right)-\widehat{F_{g,k}}\left(S'\cup\{i_e\}\right)-k\\ \Rightarrow f\left(S'\cup\{i\}\right)+\widehat{F_{g,k}}\left(S'\cup\{i\}\right)<k+\widehat{F_{g,k}}\left(S'\cup\{i_e\}\right)\\ \Rightarrow\widehat{F_{f,k}}\left(S'\cup\{i\}\right)+\widehat{F_{g,k}}\left(S'\cup\{i\}\right)<\widehat{F_{f,k}}\left(S'\cup\{i_e\}\right)+\\ \widehat{F_{g,k}}\left(S'\cup\{i_e\}\right)\\ \Rightarrow\overline{F_k}\left(S'\cup\{i\}\right)-\overline{F_k}\left(S'\right)<\overline{F_k}\left(S'\cup\{i_e\}\right)-\overline{F_k}\left(S'\right)\end{gathered}$$

(3) $f\left(S'\cup\{i\}\right)\ge k,\ f\left(S'\cup\{i_e\}\right)\ge k$.

In this case, let $b_i>p'_{i_e}$. We have:

$$b_i>f\left(S'\right)+v\left(\{i\}\right)-k \tag{10}$$

$$b_i>\widehat{F_{g,k}}\left(S'\cup\{i\}\right)-\widehat{F_{g,k}}\left(S'\cup\{i_e\}\right)-k \tag{11}$$

From inequation (10), we have

$$\widehat{F_{f,k}}\left(S'\cup\{i\}\right)=f\left(S'\cup\{i\}\right)<k \tag{12}$$

From inequation (11), we have

$$f\left(S'\cup\{i\}\right)+\widehat{F_{g,k}}\left(S'\cup\{i\}\right)<k+\widehat{F_{g,k}}\left(S'\cup\{i_e\}\right) \tag{13}$$

Combine inequation (12) and inequation (13), we have

$$\begin{gathered}\widehat{F_{f,k}}\left(S'\cup\{i\}\right)+\widehat{F_{g,k}}\left(S'\cup\{i\}\right)<\widehat{F_{f,k}}\left(S'\cup\{i_e\}\right)+\\ \widehat{F_{g,k}}\left(S'\cup\{i_e\}\right)\\ \Rightarrow\overline{F_k}\left(S'\cup\{i\}\right)-\overline{F_k}\left(S'\right)<\overline{F_k}\left(S'\cup\{i_e\}\right)-\overline{F_k}\left(S'\right)\end{gathered}$$

(4) $\left(S'\cup\{i\}\right)\ge k,\ f\left(S'\cup\{i_e\}\right)<k$.

In this case, let $b_i>p'_{i_e}$. We have:

$$b_i>f\left(S'\right)+v\left(\{i\}\right)-k, \tag{14}$$

$$\begin{gathered}b_i>v\left(\{i\}\right)-v\left(\{i_e\}\right)\\ +\widehat{F_{g,k}}\left(S'\cup\{i\}\right)-\widehat{F_{g,k}}\left(S'\cup\{i_e\}\right)+b_{i_e}\end{gathered} \tag{15}$$

From inequation (14), we have

$$\widehat{F_{f,k}}\left(S'\cup\{i\}\right)=f\left(S'\cup\{i\}\right)<k \tag{16}$$

From inequation (15), we have

$$\begin{gathered}f\left(S'\cup\{i\}\right)+\widehat{F_{g,k}}\left(S'\cup\{i\}\right)\\ <f\left(S'\cup\{i_e\}\right)+\widehat{F_{g,k}}\left(S'\cup\{i_e\}\right)\end{gathered} \tag{17}$$

Combine inequation (16) and inequation (17), we have

$$\begin{gathered}\widehat{F_{f,k}}\left(S'\cup\{i\}\right)+\widehat{F_{g,k}}\left(S'\cup\{i\}\right)<\widehat{F_{f,k}}\left(S'\cup\{i_e\}\right)+\\ \widehat{F_{g,k}}\left(S'\cup\{i_e\}\right)\\ \Rightarrow\overline{F_k}\left(S'\cup\{i\}\right)-\overline{F_k}\left(S'\right)<\overline{F_k}\left(S'\cup\{i_e\}\right)-\overline{F_k}\left(S'\right)\end{gathered}$$

To sum up, if user i bids $b_i\ge p'_{i_e}$ for all $e\in\{1,\ldots,L\}$, it will be placed after L. Hence, user i would not win the auction because the first L users have replaced it. ■

Lemma 4. *MRUS is* $(1-\epsilon)$ *-approximation to the following problem:*

$$\max\nolimits_{S\subseteq U}\min\left\{f\left(S\right),g\left(S\right)\right\},\ \ s.t.\ \ |S|\le\frac{m}{\alpha} \tag{P6}$$

where $\alpha=1+\log\left(\max_{e\in U}\left(f\left(\{e\}\right)+g\left(\{e\}\right)\right)\right)$.

Proof: Based on Theorem 4, the greedy algorithm used in *MRUS* is α − approximation to P3. Thus, we can obtain the optimal solution of P6 using binary search with $\epsilon=0$ theoretically. Consider that S^* is the optimal solution of P6, and S is the output of *MRUS*. Obviously, we have $\min\left\{f\left(S^*\right),g\left(S^*\right)\right\}\le\min\left\{f\left(U\right),g\left(U\right)\right\}=k_{max}$ and $\min\left\{f\left(S\right),g\left(S\right)\right\}\ge k_{min}$. The binary search terminates when $\left(k_{max}-k_{min}\right)<\epsilon$. Thus, we have $\min\left\{f\left(S\right),g\left(S\right)\right\}\ge\left(1-\epsilon\right)\min\left\{f\left(S^*\right),g\left(S^*\right)\right\}$. ■

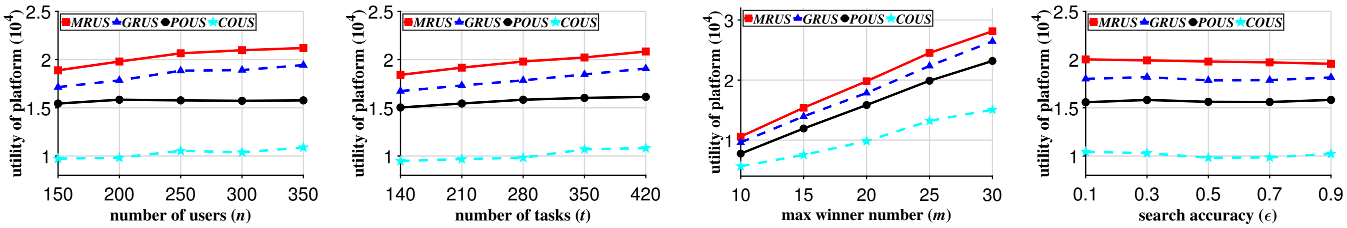
The above four lemmas together prove the following theorem.

Theorem 6. *MRUS is computationally efficient, individually rational, truthful, and* $(1-\epsilon)$ *-approximation to P6.*

VI. Performance Evaluation

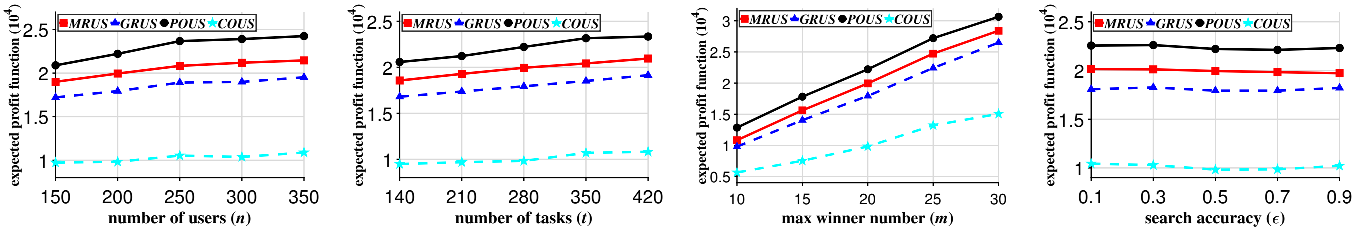
We have conducted simulations to evaluate the performance of *MRUS* based on real experience data against the following algorithms:

- *GRUS (Greedy Robust User Selection)*: This algorithm selects winners greedily to maximize the utility of the platform under the constrained number of winners.



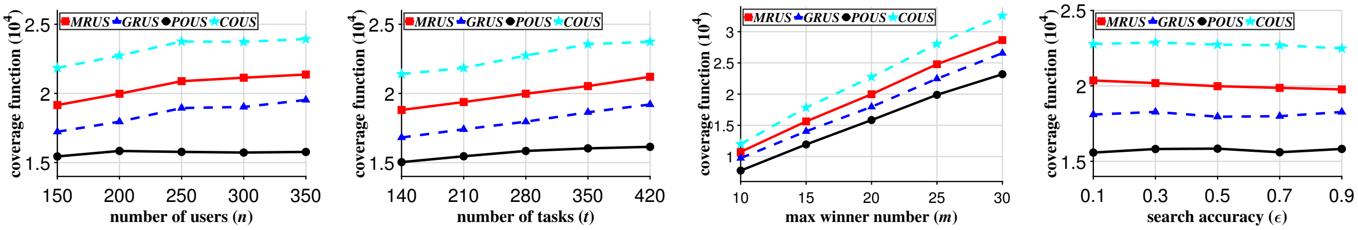
(a) (b) (c) (d)

Fig. 3: Utility of platform. (a) utility of platform versus number of users. (b) utility of platform versus number of tasks. (c) utility of platform versus maximum number of winners. (d) utility of platform versus search accuracy.



(a) (b) (c) (d)

Fig. 4: Expected profit function. (a) expected profit function versus number of users. (b) expected profit function versus number of tasks. (c) expected profit function versus maximum number of winners. (d) expected profit function versus search accuracy.



(a) (b) (c) (d)

Fig. 5: Coverage function. (a) coverage function versus number of users. (b) coverage function versus number of tasks. (c) coverage function versus maximum number of winners. (d) coverage function versus search accuracy.

- *POUS (Profit Optimization User Selection)*: This algorithm maximizes the expected profit under the constrained number of winners using *Msensing Auction* [43].
- *COUS (Coverage Optimization User Selection)*: This algorithm maximizes the coverage function under the constrained number of winners using submodular function maximization [18].

Note that there is no critical value of payment for either *GRUS* or *COUS*.

We first measure the utility of platform with different number of users (n), number of tasks (t), maximum number of winners (m) and search accuracy (ϵ). Then we measure the running time and optimality gaps, and verify the truthfulness of *MRUS*. All the simulations are run on a Windows machine with Intel(R) Core(TM) i7-7560U CPU and 16 GB memory. Each measurement is averaged over 100 instances.

A. Simulation Setup

We use the air pollution data [44] from the sites in Beijing and the *T-Drive trajectory data sample* [45], which contains trajectories of 10,357 taxis in Beijing, with geographic coordinates at different time for every trajectory. We randomly choose the sites and the taxis as the tasks and users, respectively.

For the coverage function, each site is regarded as an area, and the weights of sites are distributed uniformly over [200, 500]. Within a specified time period, a user can collect data if it is close to the site within 200m. If a user passes multiple sites, it can perform all these tasks of data collection in this time period. For our simulations, we use the taxi traces in [10:00, 15:00] at the time snapshot in 2008-02-02. The bidding price of users is selected randomly from the auction dataset [46], which contains 5017 bid prices for Palm Pilot M515 PDA from eBay. The values of tasks are distributed uniformly over [200, 450]. We set $n = 200$, $t = 280$, $m = 20$, $\epsilon = 0.5$ as the default setting. We will vary the values of the key parameters to explore the impacts on designed algorithms.

Recall that we aim to optimize the bi-objective problem. Thus the normal coefficient γ has great impact on the performance of bi-objective algorithms (*MRUS* and *POUS*). Although the slight adjustment of normal coefficient reflects the preference of the platform over the two objectives, an improper value of normal coefficient may make the bi-objective problem meaningless. In our simulations, we determine the value of normal coefficient through multiple random samplings. We select m users whose values are larger than their bidding prices as the winners. Then we calculate the value of γ such that the value of expected profit function is equal to that of coverage function. We repeat this sampling multiple times, and take the average value as the value of normal coefficient.

Actually, finding the normal coefficient is not the only way for normalization. We will discuss an alternative parameter-free normalization method, called dimensionless scale normalization, in discussion section.

B. Utility of Platform

First of all, we measure the utility of the platform of all 4 algorithms. The results are shown in Fig.3.

We vary the number of users from 150 to 350, and find that the utility of platform of *MRUS* and *GRUS* increases as the number of users increases. This is because the bi-objective algorithms have more options over a larger user set. However, since *POUS* and *COUS* only optimize a single objective, the utility of platform does not change much even if the result of their corresponding objective functions becomes better.

We vary the number of tasks from 140 to 420. The effect of the number of tasks on platform's utility of *MRUS* and *GRUS* is almost the same as that of the number of users. With more tasks to be performed, the platform can select a limited number of winners to perform the tasks, which can bring high utility to the platform. The utility of platform of *POUS* and *COUS* increases slightly. This is because when the number of sites increases, each taxi can perform more tasks averagely, and more contributions for both expected profit function and coverage function are made by each user, though either *POUS* or *COUS* only optimizes the single objective.

Then, we vary the maximum number of winners from 10 to 30. With more winners, the utility of platform of all four algorithms increases dramatically since every winner will bring the positive utility to the platform.

We vary the search accuracy from 0.1 to 0.9, and find that the outcome of *MRUS* becomes worse with the increase of search accuracy. Actually, the smaller the search accuracy is, the more precise *MRUS* is. However, the search accuracy has impact on the running time, which will be discussed later. The utilities of platform of other benchmark algorithms remain stable with different search accuracy since they do not use binary search.

Overall, the utility of the platform of *MRUS* is greater than those of other three algorithms. Specifically, *MRUS* can improve the utility by 88%, 27%, and 11% on average, compared with *COUS*, *POUS*, and *GRUS* under the default setting of our simulations, respectively.

The performance of *GRUS* is close to that of *MRUS*. However, we do not think *GRUS* is a good incentive mechanism. First, *GRUS* may work arbitrarily badly for some cases, such as the example illustrated in Fig.1. Our simulations show that *MRUS* outperforms *GRUS* in more than 76% of all cases. Thus, the generalization ability of *GRUS* is weak. Moreover, there is no truthful payment rule for *GRUS*. This means *GRUS* is not strategy-proof.

C. Expected Profit and Coverage

Since the essential goals of mobile crowdsensing systems in this paper are maximizing the expected profit and coverage, respectively, we measure the values of expected profit function and coverage function of all 4 algorithms. The results are shown in Fig.4 and Fig.5, respectively.

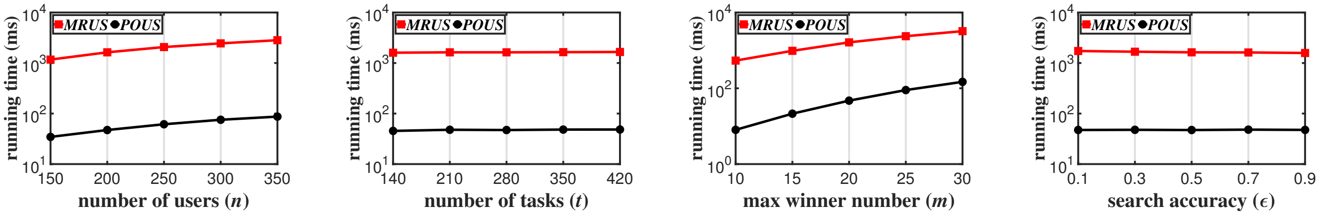
As shown in Fig.4, the performance of *POUS* on expected profit is the best since it only maximizes expected profit function. For *MRUS*, *GRUS* and *POUS*, the expected profit increases with the increasing number of users, number of tasks and maximum number of winners. The expected profit of *COUS* does not change much with the increase of number of users and number of tasks as the winners are selected only based on the contribution to coverage function in *COUS*. The value of expected profit function of *MRUS* does not change much with the increase of search accuracy. This is because the binary search is employed to maximize the minimum value of expected profit function and coverage function. Overall, *MRUS* can obtain 90% expected profit of *POUS*, and can improve the expected profit by 103% and 11% on average, compared with *COUS* and *GRUS* under the default setting of our simulations, respectively.

We can see from Fig.5 that *COUS* outperforms other algorithms in terms of coverage. For *MRUS*, *GRUS* and *COUS*, the value of coverage function increases with the increase of number of users, number of tasks and maximum number of winners. The trend of coverage function of *POUS* is uncertain with the increase of number of users as its winners are selected to maxmize the expected profit function. The value of coverage function of *POUS* increases slightly since each taxi can contribute more for both expected profit function and coverage function, averagely, though *POUS* only optimizes the expected profit function. The value of coverage function of *MRUS* does not change much with the increase of search accuracy. Overall, *MRUS* can obtain 88% coverage of *COUS*, and can improve the coverage by 26% and 11% on average, compared with *POUS* and *GRUS* under the default setting of our simulations, respectively.

D. Running Time

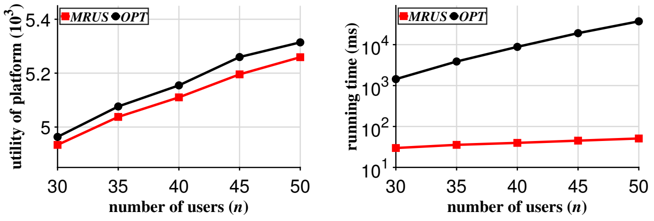
Then, we test the running time of MRUS and POUS. We do not compare the running time of *MRUS* against *GRUS* and *COUS*, since there is no truthful payment rule for either *GRUS* or *COUS*.

It can be seen from Fig.6 that the running time of both *MRUS* and *POUS* increase with the number of users, number of tasks, and maximum number of winners. Specifically, for *MRUS*, both the number of tasks and the maximum number of winners have the positive impact on value of $\min\left(f\left(U\right), g\left(U\right)\right)$. Consider that the running time of *MRUS* is $O\left(\max\left\{n^2 log\frac{\min(f(U),g(U))}{\epsilon}, n^3\right\}\right)$, which has been given in Lemma 1, the running time of *MRUS* increases. However,



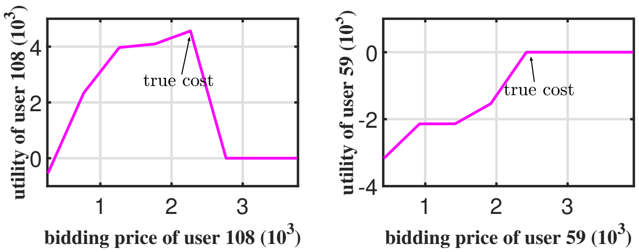
(a) (b) (c) (d)

Fig. 6: Running time. (a) running time versus number of users. (b) running time versus number of tasks. (c) running time versus maximum number of winners. (d) running time versus search accuracy.



(a) (b)

Fig. 7: Comparison with optimal solution. (a) utility of platform. (b) running time.



(a) (b)

Fig. 8: Truthfulness of *MRUS*. (a) utility of user 108 (winner). (b) utility of user ID=59 (loser).

the impact of number of tasks is very small since the values of tasks are uniformly distributed. From Fig.6(d), we can see that the running time of *MRUS* decreases slightly with the increase of search accuracy. While the running time of *POUS* does not change since it does not use binary search. The running time of *MRUS* is much larger than that of *POUS*. This is because the running time of *MRUS* largely depends on the value of $\min(f(U), g(U))$. In our simulation setting, the values of both expected profit function and coverage function are at the level of 10^4. Even so, *MRUS* can be terminated within 1.7 second under the default setting of our simulations. Moreover, the performance of *MRUS* is much better than that of *POUS*.

E. Optimality Gaps

We compare *MRUS* with the optimal solution of the original *RUS* problem in (P1) under small-scale simulations with $t = 280$, $m = 6$, $\epsilon = 0.1$. The optimal solution is implemented by enumerating all possible situations. We can see from Fig.7(a) that the utility of platform of *MRUS* is 99.1% of optimal solution in our simulations averagely, therefore, the performance of *MRUS* is very close to the optimal solution. On the other hand, as shown in Fig.7(b), *MRUS* is much faster than the optimal solution.

F. Truthfulness

We verify the cost-truthfulness of *MRUS* by randomly picking a winning user (ID=108) and a losing user (ID=59), and allowing them to bid prices that are different from their true costs. We illustrate the results in Fig.8. We can see that user 108 achieves its optimal utility if it bids truthfully ($b_{108} = c_{108} = 2266$) in Fig.8(a), and user 59 achieves its optimal utility if it bids truthfully ($b_{59} = c_{59} = 2420$) in Fig.8(b).

VII. Discussion

A. Extension to Multi-objective Problem

Our *MRUS* can be easily extended to multi-objective robust mobile crowdsensing systems, and all desirable properties still hold.

Given multiple monotonic submodular functions: $f_1(S), f_2(S), \dots, f_q(S)$, let

$$\widehat{F_{f_i,k}}(S) = \min\{f_i(S), k\}, i = 1, 2, \dots, q \tag{18}$$

$$\overline{F_k}(S) = \frac{1}{q}\sum_{i=1,2,\dots,q}\widehat{F_{f_i,k}}(S) \tag{19}$$

The extended *MRUS* is $(1-\epsilon)$ -approximation to the following problem:

$$\max_{S\subseteq U}\ \min_{i=1,2,\dots,q}\{f_i(S)\},\ \ s.t.\ |S| \le \frac{m}{\alpha} \tag{P7}$$

where $\alpha = 1 + \log\left(\max_{e\in U}\sum_{i=1,2,\dots,q} f_i(\{e\})\right)$.

Moreover, if at least one function is bidding price related, the critical value can be determined. Thus we can still design the multi-objective robust incentive mechanism based on the auction, and all desirable properties of *MRUS* still hold:

Theorem 7. *The extended MRUS is computationally efficient, individually rational, truthful, and* $(1-\epsilon)$ *-approximation to P7.*

For the general multi-objective problem with non-submodular functions, it is hard to obtain the approximate solution. However, many methods have been proposed to obtain the *Pareto optimal* [39] solutions.

B. Dimensionless Scale Normalization

In *MRUS*, we need to determine the value of normal coefficient carefully since the solution is sensitive to the normalization of the objective functions. Although we have proposed multiple random samplings to determine the value of normal coefficient, it is hard to say multiple random samplings can normalize the functions completely.

Another normalization way is dimensionless scale normalization, which normalizes the objectives into a uniform and dimensionless scale. Let

$$g'(S) = \sum_{l \in Z} w_l \cdot \log\left(1 + n_l(S)\right) \tag{20}$$

The normalized expected profit function is defined as:

$$f_{norm}(S) = \frac{f(S)}{f(U)} \tag{21}$$

The normalized coverage function is defined as:

$$g_{norm}(S) = \frac{g'(S)}{g'(U)} \tag{22}$$

$f_{norm}(S)$ and $g_{norm}(S)$ can be viewed as the degree of satisfaction of expected profit and coverage, respectively. $f(U)$ and $g'(U)$ are the maximum expected profit and coverage can be achieved, respectively. Note that, given user set U, either $f(U)$ or $g'(U)$ is a constant.

Then the *RUS* problem for the normalized functions can be formulated as follows:

$$\max_{S \subseteq U} \min\left\{f_{norm}(S), g_{norm}(S)\right\}, \quad s.t. \quad |S| \le m \tag{P8}$$

Theorem 8. *Both normalized expected profit function f_{norm} and normalized coverage function g_{norm} are nonnegative, monotone, and submodular functions.*

Proof: We first show that f_{norm} is a nonnegative, monotone, and submodular function. The nonnegativity of f_{norm} is obvious. The monotonicity of f_{norm} is also obvious as adding a new user into S cannot decrease the value of f_{norm}.

For all $A \subseteq B \subseteq U$ and $e \in U \backslash B$, we have

$$\begin{aligned} &f_{norm}(A \cup \{e\}) - f_{norm}(A) \\ &= \frac{f(A \cup \{e\})}{f(U)} - \frac{f(A)}{f(U)} \\ &\ge \frac{f(B \cup \{e\})}{f(U)} - \frac{f(B)}{f(U)} \\ &= f_{norm}(B \cup \{e\}) - f_{norm}(B) \end{aligned}$$

where the inequation relies on the submodularity of function f.

Thus, f_{norm} is submodular.

Similarly, we can obtain that g_{norm} is also nonnegative, monotone, and submodular. ■

Since the normalized functions are nonnegative, monotone, and submodular functions, our *MRUS* can be applied to solve P8, and all desirable properties still hold.

C. Adaption to Online Scenarios

In this subsection, we discuss how can *MRUS* adapt to online scenarios.

If the users arrive online, the online auction algorithm frame using multiple-stage sampling-accepting process [24] can be employed. Consider that the total time to recruit users is T. We divide T into $\lfloor log_2 T \rfloor + 1$ stages: $\left\{1, 2, \ldots, \left\lfloor \log_2 T \right\rfloor, \left\lfloor \log_2 T \right\rfloor + 1\right\}$. The stage i ends at time step $T' = \left\lfloor 2^{i-1} T / 2^{\lfloor \log_2 T \rfloor} \right\rfloor$. Correspondingly, the number of winners of stage i is not more than $\left\lfloor 2^{i-1} m / 2^{\lfloor \log_2 m \rfloor} \right\rfloor$. We execute *MRUS* at each stage. The difference is that we should select a user as the winner only if its marginal density is not less than a certain density threshold computed using previous users' information. In our system model, the density threshold can be calculated as the average value of $\overline{F_k}$ in the previous stages. At every stage, the user is selected as the winner only if the marginal contribution to $\overline{F_k}$ is larger than the density threshold. Since function $\overline{F_k}$ is monotone and submodular, such online auction can satisfy the desirable properties of individual rationality, cost-truthfulness, and time-truthfulness.

If both tasks and users arrive online, the above online auction is invalid since it is hard to determine the number of winners of every stage. This is because the performable tasks are uncertain at each stage. In this scenario, one viable and straightforward solution is to divide the time into multiple time intervals, and execute *MRUS* based on the available tasks and users at each time interval. Essentially, such approach transforms the online scenario into multiple offline scenarios. Moreover, the normal coefficient can be updated according to the knowledge from previous time intervals.

VIII. Conclusion

In this paper, we have optimized the worst performance of a bi-objective problem in mobile crowdsensing to improve the system robustness. We have modeled an auction-based bi-objective robust mobile crowdsensing system, and designed two independent objective functions to maximize the expected profit and coverage, respectively. We have shown that both functions of expected profit and coverage are nonnegative, monotone, and submodular. We have formulated the *RUS* problem, and designed an incentive mechanism, which utilizes the combination of binary search and greedy algorithm, to solve the *RUS* problem. We have demonstrated that the designed incentive mechanism, *MRUS*, satisfies the desirable properties of computational efficiency, individual rationality, truthfulness, and constant approximation to the tightened *RUS* problem. We have shown that our *MRUS* can be easily extended to multi-objective robust mobile crowdsensing systems. Moreover, our incentive mechanism can achieve 11% improvement of platform's utility compared with the greedy algorithm for bi-objective mobile crowdsensing systems on average.

References

[1] J. Lee, and B. Hoh, "Sell your experiences: A market mechanism based incentive for participatory sensing," in *Proc. of IEEE PerCom*, 2010, pp. 60-68.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3072953, IEEE Internet of Things Journal

[2] I. Krontiris, and A. Albers, “Monetary incentives in participatory sensing using multi-attributive auctions,” *Journal of Parallel, Emergent and Distributed Systems*, vol. 27, no. 4, pp. 317-336, 2012.

[3] L. Gao, F. Hou, and J. Huang, “Providing long-term participation incentive in participatory sensing,” in *Proc. of IEEE INFOCOM*, 2015, pp. 2803-2811.

[4] H. Jin, B. He, L. Su, K. Narstedt, and X. Wang, "Data-Driven Pricing for Sensing Effort Elicitation in Mobile Crowd Sensing Systems", *IEEE/ACM Transactions on Networking*, vol. 27, no. 6, pp. 2208-2221, 2019.

[5] J. Xu, C. Guan, H. Wu, D. Yang, L. Xu, and T. Li, “Online incentive mechanism for mobile crowdsourcing based on two-tiered social crowdsourcing architecture,” in *Proc. of IEEE SECON*, 2018, pp. 1–9.

[6] L. Jiang, X. Niu, J. Xu, D. Yang, and L. Xu, “Incentivizing the workers for truth discovery in crowdsourcing with copiers,” in *Proc. of IEEE ICDCS*, 2019, pp. 1286–1295.

[7] J. Lee, and B. Hoh, “Dynamic pricing incentive for participatory sensing,” *Pervasive and Mobile Computing*, vol. 6, no. 6, pp. 693-708, 2010.

[8] J. Xu, J. Xiang, and D. Yang, “Incentive mechanisms for time window dependent tasks in mobile crowdsensing,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6353-6364, 2015.

[9] W. Nan, B. Guo, S. Huangfu, Z. Yu, H. Chen, and X. Zhou, “A cross-space, multi-interaction-based dynamic incentive mechanism for mobile crowdsensing,” in *Proc. of IEEE Ubiquitous Intelligence and Computing*, 2014, pp. 179-186.

[10] K. Han, C. Zhang, and J. Luo, “Taming the uncertainty: Budget limited robust crowdsensing through online learning,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1462-1475, 2016.

[11] R. Rana, C. Chou, S. Kanhere, N. Bulusu, and W. Hu, “Earphone: An end-to-end participatory urban noise mapping,” in *Proc. of ACM/IEEE IPSN*, Apr. 2010, pp. 105–116.

[12] J. Carrapetta, N. Youdale, A. Chow, and V. Sivaraman, Haze Watch Project. [Online]. Available: http://www.hazewatch.unsw.edu.au/

[13] E. Koukoumidis, L. Peh, and M. Martonosi, “SignalGuru: Leveraging Mobile Phones for Collaborative Traffic Signal Schedule Advisory,” in *Proc. of ACM MobiSys*, 2011, Bethesda, pp. 127-140.

[14] Y. Liu, Y. Zhao, L. Chen, J. Pei, and J. Han, “Mining frequent trajectory patterns for activity monitoring using radio frequency tag arrays,” *IEEE Transactions on Parallel & Distributed Systems*, vol. 23, no. 11, pp. 2138–2149, 2011.

[15] H. Jin, L. Su, D. Chen, K. Nahrstedt, and J. Xu, “Quality of information aware incentive mechanisms for mobile crowd sensing systems,” in *Proc. of ACM MobiHoc*, 2015, pp. 167–176.

[16] D. Peng, F. Wu, and G. Chen, “Pay as how well you do: A quality based incentive mechanism for crowdsensing,” in *Proc. of ACM MobiHoc*, 2015, pp. 177–186.

[17] H. Jin, L. Su, D. Chen, H. Guo, K. Nahrstedt, and J. Xu, "Thanos: Incentive Mechanism with Quality Awareness for Mobile Crowd Sensing", *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1951-1964, 2019.

[18] G. L. Nemhauser, L. A. Wolsey and M. L. Fisher, “An analysis of approximations for maximizing submodular set functions I,” *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[19] L. Blumrosen and N. Nisan, Combinatorial auctions (a survey). In N. Nisan, T. Roughgarden, E. Tardos and V. Vazirani, editors, “Algorithmic Game Theory,” *Cambridge University Press*, 2007.

[20] L. G. Jaimes, I. Vergara-Laurens, and M. A. Labrador, “A location-based incentive mechanism for participatory sensing systems with budget constraints,” in *Proc. of IEEE Pervasive Computing and Communications*, 2012, pp. 103-108.

[21] 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 *Proc. of IEEE INFOCOM*, 2014, pp. 1231-1239.

[22] D. Zhang, H. Xiong, L. Wang, and G. Chen, “CrowdRecruiter. Selecting participants for piggyback crowdsensing under probabilistic coverage constraint,” in *Proc. of ACM UbiComp*, 2014, pp. 703-714.

[23] J. Xu, C. Guan, H. Dai, D. Yang, L. Xu, and J. Kai, “ Incentive Mechanisms for Spatio-temporal Tasks in Mobile Crowdsensing,” in *Proc. of IEEE MASS*, 2019, pp. 4-7.

[24] D. Zhao, X. Li, and H. Ma, “How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint,” in *Proc. IEEE INFOCOM*, 2014, pp.1213-1221.

[25] J. Lin, M. Li, D. Yang, G. Xue and J Tang, “Sybil-proof incentive mechanisms for crowdsensing,” in *Proc. of IEEE INFOCOM*, 2017, pp. 1-9.

[26] J. Lin, M. Li, D. Yang and G. Xue, “Sybil-proof online incentive mechanisms for crowdsensing,” in *Proc. of IEEE INFOCOM*, 2018, pp. 2438-2446.

[27] Y. Zhang, X. Zhang, and F. Li, “BiCrowd: Online Biobjective Incentive Mechanism for Mobile Crowdsensing,” *IEEE Internet of Things Journal*, vol. 7, no.11, pp.11078-11091, 2020.

[28] H. Gao, C. Liu, J. Tang, D. Yang, P. Hui, and W.Wang, (2018). “Online quality-aware incentive mechanism for mobile crowd sensing with extra bonus,” *IEEE Transactions on Mobile Computing*, vol. 18, no.11, pp. 2589-2603, 2018.

[29] K. Han, C. Zhang, and J. Luo, “BLISS: Budget limited robust crowdsensing through online learning,” in *Proc. of IEEE SECON*, 2014, pp. 555-563.

[30] K. Han, H. Huang, and J. Luo, “Quality-aware pricing for mobile crowdsensing,” *IEEE/ACM Transactions on Networking*, vol. 26, no. 4, pp. 1728-1741, 2018.

[31] K. Han, H. Huang, and J. Luo, “Posted pricing for robust crowdsensing,” in *Proc. of ACM MobiHoc*, 2016, pp. 261-270.

[32] Y. Qu, S. Tang, C. Dong, P. Li, S. Guo, H. Dai, and F. Wu, “Posted pricing for chance constrained robust crowdsensing,” *IEEE Transactions on Mobile Computing*, vol.19, no. 1, pp. 188-199, 2018.

[33] J. Xu, H. Li, Y. Li, D. Yang, and T. Li, “Incentivizing the biased requesters: Truthful task assignment mechanisms in crowdsourcing,” in *Proc. of IEEE SECON*, 2017, pp. 1-9.

[34] J. Xu, Z. Rao, L. Xu, D. Yang, and T. Li, “Incentive mechanism for multiple cooperative tasks with compatible users in mobile crowd sensing via online communities,” *IEEE Transactions on Mobile Computing*, vol. 19, no. 7, pp. 1618-1633, 2020.

[35] K. Deb, “Multi-objective Optimization,” *Search methodologies. Springer*, pp. 403-449, 2014.

[36] S. Chanas, “Fuzzy programming in multiobjective linear programming—a parametric approach,” *Fuzzy sets and Systems*, vol. 29, no. 3, pp. 303-313, 1989.

[37] K. Deb, “Multi-objective optimization using evolutionary algorithms,” *John Wiley & Sons*, 2001.

[38] M. Reyes-Sierra, and C. Coello, “A Multi-objective particle swarm optimizers: A survey of the state-of-the-art,” *International journal of computational intelligence research*, vol.2, no.3, pp. 287-308, 2006.

[39] A. Sen, “Markets and freedoms: achievements and limitations of the market mechanism in promoting individual freedoms,” *Oxford Economic Papers*, pp. 519-541, 1993.

[40] U. Feige, “A threshold of ln n for approximating set cover,” *Journal of the ACM*, vol. 45, no. 4, pp. 634-652, 1998.

[41] L. Wolsey, “An analysis of the greedy algorithm for the submodular set covering problem,” *Combinatorica*, vol. 2, no. 4, pp.385-393,1982.

[42] R. Myerson, “Optimal auction design,” *Mathematics of operations research*, vol. 6, no. 1, pp. 58-73, 1981.

[43] D. Yang, G. Xue, X. Fang and J. Tang, “Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing,” in *Proc. of ACM Mobicom*, 2012, pp. 173-184.

[44] http://beijingair.sinaapp.com

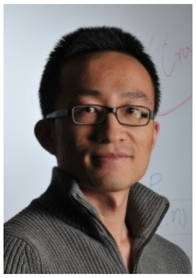
[45] https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample

[46] http://www.modelingonlineauctions.com/datasets

Jia Xu (*M'15*) received the M.S. degree in School of Information and Engineering from Yangzhou University, Jiangsu, China, in 2006 and the PhD. Degree in School of Computer Science and Engineering from Nanjing University of Science and Technology, Jiangsu, China, in 2010. He is currently a professor in the School of Computer Science at Nanjing University of Posts and Telecommunications. He was a visiting Scholar in the Department of Electrical Engineering & Computer Science at Colorado School of Mines from Nov. 2014 to May. 2015. His main research interests include crowdsourcing, edge computing and wireless sensor networks. Prof. Xu has served as the PC Co-Chair of SciSec 2019, Organizing Chair of ISKE 2017, and TPC member of Globecom, ICC, MASS, ICNC, EDGE. He currently serves as the Publicity Co-Chair of SciSec 2021.

Yuanhang Zhou is now pursing the bachelor degree in Bell Honors School, Nanjing University of Posts and Telecommunications, Jiangsu, China. His research interests are mainly in the areas of the mobile crowdsensing, machine learning, and game theory.

Yuqing Ding is now pursing the bachelor degree in Bell Honors School, Nanjing University of Posts and Telecommunications, Jiangsu, China. His research interests include mobile crowdsensing, machine learning, and graph theory algorithms.



Dejun Yang (*SM'19*) received the B.S. degree in computer science from Peking University, Beijing, China, in 2007 and the Ph.D. degree in computer science from Arizona State University, Tempe, AZ, USA, in 2013. Currently, he is an associate professor of computer science with Colorado School of Mines, Golden, CO, USA. His research interests include Internet of things, networking, and mobile sensing and computing with a focus on the application of game theory, optimization, algorithm design, and machine learning to resource allocation, security and privacy problems. Prof. Yang has served as the TPC Vice-Chair for Information Systems for IEEE International Conference on Computer Communications (INFOCOM) and currently serves an associate editor for the IEEE Internet of Things Journal (IoT-J). He has received the IEEE Communications Society William R. Bennett Prize in 2019 (best paper award for IEEE/ACM Transactions on Networking (TON) and IEEE Transactions on Network and Service Management in the previous three years), Best Paper Awards at IEEE Global Communications Conference (GLOBECOM) (2015), IEEE International Conference on Mobile Ad hoc and Sensor Systems (2011), and IEEE International Conference on Communications (ICC) (2011 and 2012), as well as a Best Paper Award Runner-up at IEEE International Conference on Network Protocols (ICNP) (2010).

Lijie Xu received his Ph.D. degree in the Department of Computer Science and Technology from Nanjing University, Nanjing, in 2014. He was a research assistant in the Department of Computing at the Hong Kong Polytechnic University, Hong Kong, from 2011 to 2012. He is currently an associate professor in the School of Computer Science at Nanjing University of Posts and Telecommunications, Nanjing. His research interests are mainly in the areas of wireless sensor networks, ad-hoc networks, mobile and distributed computing, and graph theory algorithms.