An Ordered Submodularity-Based Budget-Feasible Mechanism for Opportunistic Mobile Crowdsensing Task Allocation and Pricing

Jixian Zhang, Yi Zhang, Hao Wu, and Weidong Li

Abstract—Mobile crowdsensing services are divided into two categories: opportunistic and participatory. In opportunistic mobile crowdsensing services, users do not need to specify the crowdsensing tasks to be completed. Compared with participatory crowdsensing services, the application scope is wider and more user-friendly. In participatory crowdsensing, the service provider assumes that the user can successfully complete the data collection task. However, such an approach cannot work in an opportunistic crowdsensing service because in opportunistic crowdsensing, the user's execution of the task is uncertain, which brings great challenges to the quality of the crowdsensing service. This article is based on the assumption of the user coverage probability model and transforms the opportunistic mobile crowdsensing value maximization problem into an ordered submodularity value function model with budget constraints. This model is also good at representing participatory crowdsourcing problems. To the best of our knowledge, this is the first study to apply the ordered submodularity feature to a mobile crowdsensing service. Furthermore, we combine the properties of ordered submodular and auction models and propose an ordered submodularity-proportional share mechanism (O-PSM) to solve the allocation and payment problems in opportunistic mobile crowdsensing services. Specifically, in the allocation stage, the winning users are selected based on the proportional share threshold, and in the payment stage, the payment price for the winning users is designed based on critical value theory. We prove that the mechanism satisfies the economic characteristics of individual rationality, truthfulness, and budget feasibility. In the experimental section, the mechanism design based on ordered submodularity is shown to enable the service provider to obtain a higher value and a lower payment.

Index Terms—Coverage probability distribution, mechanism design, opportunistic mobile crowdsensing, ordered submodularity

1 Introduction

Mobile crowdsensing services (MCSs) are an emerging business model that is mostly used by service providers to collect data of interest through mobile users. For example, the environmental department collects environmental noise data through users' mobile device microphones, the mobile operator collects the signal strength through the mobile devices, the transportation department collects traffic flow information through the mobile device GPS [1], and blockchain-based MCSs use majority-voting schemes to determine the outcomes for participants [2]. The early crowdsensing services mostly adopted a volunteer method, and users completed the tasks released by the service providers for free. However, this method has low user participation and cannot guarantee the quality of the crowdsensing services. To encourage more users to participate in crowdsensing services, service providers adopt incentive mechanisms [3], such as paying a certain fee to the users who complete the tasks. We call this a crowdsensing service with auction models.

- *Jixian Zhang, Yi Zhang, and Hao Wu are with the School of Information Science and Engineering, Yunnan University, Kunming 650504, China. E-mail: denonji@163.com, yamizy@mail.ynu.edu.cn, haowu@ynu.edu.cn.*
- *Weidong Li is with the School of Mathematics and Statistics, Yunnan University, Kunming 650504, China. E-mail: weidongmath@126.com.*

Manuscript received 1 September 2022; revised 17 November 2022; accepted 23 December 2022. Date of publication 27 December 2022; date of current version 8 January 2024.
This work was supported in part by the National Natural Science Foundation of China under Grants 62062065, 12071417, 61962061, in part by the Natural Science Foundation of Yunnan Province of China under Grant 2018ZF017, in part by a major projects of Yunnan Province of China under Grant 202202AD080007, in part by the Education Foundation of Yunnan Province of China under Grant 2022J002 and the Program for Excellent Young Talents, Yunnan, China.
(Corresponding author: Weidong Li.)
Digital Object Identifier no. 10.1109/TMC.2022.3232513

1.1 Motivation

Literature [4] classifies MCSs into *opportunistic* and *participatory* types. For participatory MCSs, users actively participate in the data collection process and can complete crowdsensing data collection services within a specified time and area. Most of the current studies [5], [6] are based on this assumption. For opportunistic MCSs [7], [8], users do not need to participate much. For example, when they reach a specified area, the device will start to perform the data collection tasks automatically. In such a crowdsensing system, users only need to confirm whether to participate and provide the expected benefits when the crowdsensing service is released. The system selects users based on historical information. Obviously, using the model of participatory MCSs to describe the problem of opportunistic crowdsensing is inappropriate. Because the data collection of opportunistic crowdsensing is uncertain, it is affected by various factors, such as user behavior and the mobile device hardware. For example, when using crowdsensing services to predict traffic flow, there is no guarantee that the users will follow a preset route. As another example, when the period of the noise detection crowdsensing service is long (1-2 months), asking the user to appear at the specified place and time is not very convenient. This makes traditional participatory MCS models face great challenges in describing opportunistic crowdsensing.

Among the representations of the numerous crowdsensing problems, the point of interest (POI)-based models are favored by many researchers. Their location-based features make them applicable to many types of crowdsensing services, such as noise detection, signal detection, traffic condition prediction and air quality detection. The basic principle is that the service provider declares multiple POIs in the data collection area [5] while simultaneously announcing the corresponding data collection task for each POI. When the user is at the POI, the mobile device can be used to collect data. Notably, as the user moves, data from multiple POIs may be collected. Applying the POI model to participatory MCSs is appropriate. For example, [5] and [6] both build a monotonic submodular value model based on the coverage times of the POIs by the users. However, applying the POI model to opportunistic MCSs brings many problems. An important shortcoming in existing research is that the service provider assumes that the user can successfully perform the data collection task at the declared POI, which is inconsistent with the reality of opportunistic crowdsensing services. For example, if the user performs a long-term noise collection task, then the quality of the service cannot be guaranteed. However, in participatory MCSs, the service provider calculates the final payment based on the declaration submitted by the user. When the actual data collection situation is inconsistent with the declaration submitted by the user, the service provider may experience considerable losses. In short, classic POI models cannot effectively represent opportunistic MCSs. Designing a suitable model for opportunistic MCSs is a major challenge.

Another problem in the opportunistic MCS problem is payment pricing. Under the premise of budget constraints, the service provider must ensure that the price paid to the winning user is greater than the user's bid, which is equivalent to a reverse auction mechanism design. Since users are selfish and may obtain greater profits by manipulating bids, the mechanism design must meet the characteristics of individual rationality, truthfulness (also known as incentive-compatible or strategy-proof) and budget feasibility. However, the features of opportunistic MCSs entail another challenge in mechanism design. Fig. 1 shows a diagram of POI-based opportunistic mobile crowdsensing service with an auction model. The basic principle is that 1) the service provider releases a crowdsensing service task in a certain area (such as collecting noise data on POIs), 2) the users accept the task and submit bids as the expected compensation for completing the data collection task, 3) the service provider selects the most valuable users according to the historical data collected (such as user activity areas), 4) the service provider pays the users a fee as compensation for completing the task, and 5) the users complete the data collection and feed the data back to the service provider. The biggest difference between opportunistic crowdsensing services and participatory crowdsensing services is that users do not have to actively inform service providers of the POIs they can reach.

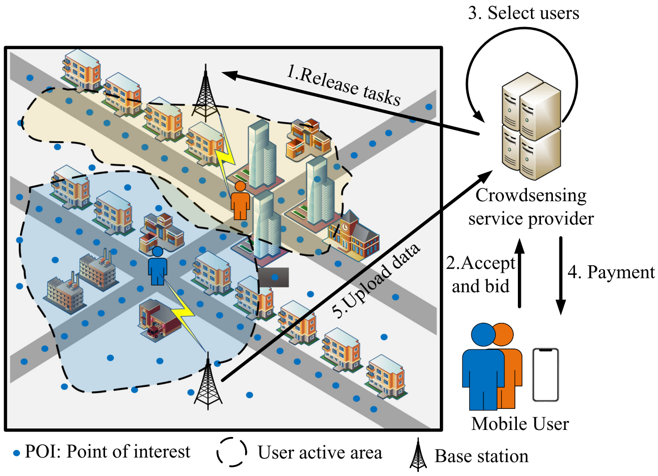


Fig. 1. POI-based opportunistic mobile crowdsensing service with auction model.

1.2 Contributions

In response to the uncertainty of opportunistic MCSs, we adopt a prior probability model to represent the user's POI coverage with a more scientific approach than the traditional 0 or 1 coverage. For example, a user has a probability of more than 90% to cover POIs in areas where she/he often visits and only a 10% probability to cover POIs in areas where she/he does not often visit. Such a model can correctly evaluate the value of users in opportunistic MCSs. However, it leads to problems for which the traditional monotonic submodular model cannot work. The crowdsensing problem has the characteristics of diminishing marginal utility, and the existing research generally adopts the submodular function to represent the value model of the crowdsensing problem. The monotonic submodular function is a set function used to find the subset with the greatest value, and the order of the users in the set does not affect the value. However, this approach is not applicable to the POI coverage problem represented by probabilistic models. For each POI constrained by the maximum number of coverage times, and users have different coverage probabilities, in the winner set, different user orders will produce different coverage values, so the traditional submodular function cannot correctly represent our model. A previous study [9] defines this type of submodular function that is affected by the order of the elements as an *ordered submodularity function*.

Many important optimization problems involve selecting a subset of items from a larger set. An implicit modeling assumption is that the order of the selected elements does not matter since submodularity is a property of functions that operate on unordered sets. However, in many applications, we care about not only the selected set of users but also the order of users. An intuitive example is that a company is composed of certain employees, but the order in which employees join the company will have a considerable impact on the company's revenue. This is also the essential difference between the classical submodularity property and the ordered submodularity property.

In this article, we study a more general opportunistic MCS problem in which the service provider declares several POIs in an area of interest and sets the maximum number of coverage times for each POI. Additionally, each user has a different coverage probability for different POIs, and the service provider selects the most valuable users to win according to the probability model and the bids submitted by the users and calculates the final payment price given to winners without exceeding the budget. Notably, our proposed model is compatible with the traditional participatory MCS problem (when the coverage probability greater than a certain threshold is set to 1). Specifically:

- This article uses prior probability to model the POI coverage problem in opportunistic MCSs and transforms the MCS value maximization problem into an ordered submodularity value function model with budget constraints. To the best of our knowledge, this is the first study to combine MCSs with ordered submodularity, which is quite instructive for the study of crowdsensing problems.
- We design a reverse auction mechanism to solve the winner decision and payment calculation problem in the MCS. Specifically, by improving the proportional share mechanism (PSM) [10], we propose an ordered submodularity-based proportional share mechanism (O-PSM) for the problem, and we prove that O-PSM satisfies individual rationality and truthfulness and has a polynomial running time.

The rest of this article is organized as follows: In Section 2, we discuss the existing studies on mobile crowdsensing services and mechanism design. In Section 3, we describe the problem of mobile crowdsensing with ordered submodularity features and mechanism design preliminaries. In Section 4, we propose an ordered submodularity PSM design for solving the MCS value maximization problem; then, we prove that the mechanism satisfies individual rationality and truthfulness. In Section 5, we evaluate the mechanisms through extensive experiments. Finally, in Section 6, we summarize our results and present possible directions for future research.

2 Related Works

Early crowdsensing services mostly adopted a volunteer method. However, this method has low user participation and cannot guarantee the quality of the crowdsensing services. To encourage more users to participate in crowdsensing services, service providers adopt incentive mechanisms [3]. Auctions are an effective strategy to encourage users to participate in market activities. Auction mechanisms have been widely used in spectrum auctions [11], [12], offline and online cloud computing virtual resource auctions [13], [14], [15], edge computing [16], [17], and mobile blockchain [18], [19]. Different from the above auction mechanisms, the mobile crowdsensing service mostly uses reverse auctions. This means that the service provider pays the user to purchase the user's collected data.

Crowdsensing services based on the auction model are divided into two types: participatory and opportunistic. Participatory crowdsensing services refer to users actively participating in the data collection process, and the crowdsensing data collection services can be completed within a specified time and area. Participatory crowdsensing services have received more attention. Singer [10] first studied the mechanism with budget constraints and designed an advanced PSM algorithm. The main result indicated that a bounded approximation ratio is achievable for an important class of submodular functions, and the use of submodular functions for mechanism designs was pioneered. Anari et al., [20] considered a mechanism design problem in the context of large-scale crowdsensing markets and designed a budget-feasible mechanism for large markets that achieves a competitive ratio of 1 - 1/e = 0.63. Based on the online auction model, Zhao et al., [5] investigated a problem in which users submit their private types to the crowdsourcer in a random order, and they designed two online mechanisms, OMZ and OMG, satisfying the computational efficiency, individual rationality, budget feasibility, truthfulness, consumer sovereignty and constant competitiveness constraints under the zero arrival-departure interval case and a more general case. Zheng et al., [6] studied the coverage problem for incentive-compatible mobile crowdsensing and proposed BEACON, which is a budget-feasible and strategy-proof incentive mechanism for weighted-coverage maximization in mobile crowdsensing, and adopted a newly designed proportional-share-rule-based compensation-determination scheme to guarantee strategy-proofness and budget feasibility. Zheng et al., [21] investigated the problem of online crowdsensing by considering the critical property that the values of the user contributions decrease as time passes. They proposed a new method to select users based on a time-related threshold and presented a strategy-proof framework. Hu et al., [22] proposed a novel blockchain-based MCS framework that preserves privacy and secures both the sensing process and the incentive mechanism by leveraging emergent blockchain technology, and they designed an incentive mechanism by means of a three-stage Stackelberg game. Han et al., [23] considered a scenario in which a mobile crowdsensing platform aims to maximize crowdsensing revenue under a budget constraint and the users are interested in maximizing their utility while keeping their cost private. Wang et al., [24] proposed an incentive mechanism to maximize the social welfare for mobile crowdsensing and designed a discrete particle swarm optimization (DPSO) algorithm for worker-centric task selection to maximize worker utility. The greatest problem for participatory crowdsensing services is that the uncertainty of the data collection reduces service quality.

Ganti et al., [4] define opportunistic MCS as "opportunistic sensing is where the sensing is more autonomous and user involvement is minimal (e.g., continuous location sampling)". Based on opportunistic MCSs, Ma et al., [25] investigated the opportunistic characteristics of human mobility from the perspective of sensing and transmission, discussed how to exploit these opportunities to collect data efficiently and effectively and proposed the concept of mobile crowdsensing data collection location. Zhan et al., [26] proposed a time-sensitive incentive-aware mechanism for mobile opportunistic crowdsensing data collection, in which each sensing data point has an attached time-sensitive value that decays over time. Zhan et al., [7] also focused on data collection in mobile opportunistic crowdsensing, where the data can be transferred between mobile users via opportunistic device-to-device communications. Yucel et al.,

[8] proposed online stable task assignment algorithms for worker trajectories that are uncertain in opportunistic mobile crowdsensing. They also studied the problem of finding task assignments that fulfill both the coverage-aware preferences of the service requesters and the profit-based preferences of the workers in a budget-constrained, opportunistic mobile crowdsensing system [27]. Although the above studies analyzed opportunistic MCSs, they were not based on the spatial coverage model and did not pay attention to the economic characteristics of the mechanism design.

Most crowdsensing services have the characteristic of diminishing marginal utility, so many participatory crowdsensing service problems will be transformed into submodular functions to be solved. The first such mechanism was developed by Dobzinski, Nisan and Schapira [28], who reported an approximation mechanism under the submodular value function model. This problem has been extended by Dobzinski [29] and Assadi et al., [30]. Because of the uncertainty brought by opportunistic crowdsensing, traditional submodular functions cannot work effectively, and the basic theory of this problem had not been solved until recently. Kleinberg et al., [9] proposed a new type of submodular function, an ordered submodularity function, where the order of the elements in the set affects the value of the function, and gave the definition, proof and basic algorithm of the ordered submodularity function. The emergence of ordered submodularity functions provides a new theoretical basis for modeling opportunistic MCSs.

Inspired by the above literature, this article applies ordered submodularity theory to model the opportunistic MCS value maximization problem for the first time and designs a budget-feasible mechanism that can achieve truthfulness, individual rationality, budget feasibility and computational efficiency.

3 Mechanism Design of Mobile Crowdsensing With Ordered Submodularity Features

In this section, we first present a probability-based POI coverage problem system model of an MCS, prove the ordered submodularity property, and then explain the preliminaries of the mechanism design with budget feasibility of the MCS.

3.1 Probability-Based POI Coverage Problem

We assume that the service provider declares M POIs in the certain area D for which data need to be collected, represented by the set $\mathcal{M} = \{1, 2, \dots M\}$. For any POI $m \in \mathcal{M}$, define $r_m \in Z^+$ as the maximum coverage times of the POI (at most r_m users are required to collect data at this POI for effective analysis). If the service provider wants to collect more valuable data, it needs to increase the r_m, which means that more users can be selected, but the expenditure of the service provider is limited by the budget. On the other hand, if the r_m is too small, it may be unable to complete the crowdsensing task. We believe that service providers will carefully evaluate the number of times each POI needs to be covered and finally be able to complete the crowdsensing task without exceeding the budget. If the data can be completely and successfully collected, then the service provider can obtain the value of $v_m \in R^+$ at this POI. In general, the service provider can define the value of the crowdsensing service data itself, depending on how the service provider understands the data collected at the POI. v_m can be understood as the value expectation obtained by the service provider after completing the data collection task at POI m. For example, for noise data collection, if the service provider is more concerned about the occurrence of noise in certain areas, then multiple sampling tests are required, so the POI coverage times can be used as the basis for judging the value obtained by the service provider. The service provider's maximum budget is B.

Moreover, there are N users participating in the crowdsensing service, which is represented by set $\mathcal{N} = \{u_1, u_2, \dots u_N\}$. For user $u_i \in \mathcal{N}$, we use i instead of u_i under the premise of no ambiguity. Due to the user's position, device hardware reasons or the user's autonomous behavior (for example, the noise data collection task cannot be completed because the microphone is blocked or the mobile phone is placed in a bag and the signal collection task cannot be completed), the user may not be able to fully complete the data collection task at POIs, which yields uncertain results in terms of the quality of the crowdsensing service. We use prior probabilities to describe the likelihood of user coverage for each POI. The prior probability comes from 2 types of information. One depends on the relationship between the POI location and the user's daily activity area. According to the analysis of the user's daily activity area, we can obtain the probability of the user appearing at each location in area D in advance; we use $pr_i(x, y) \in [0, 1], (x, y) \in D$ to represent this value. This probability can be simply understood as the time the user remains at location (x, y) during a period of time. For example, during a day, if the user remains in their bedroom for 8 hours, then the probability of the user appearing in the bedroom is $1/3$. Clearly, $\iint_{(x,y)\in D} pr_i(x, y)dxdy \in [0, 1]$. Moreover, we define the distance factor $e^{-\sqrt{(x-x_m)^2+(y-y_m)^2}}$ between position (x, y) and POI m center (x_m, y_m). If the user's active area is very close to the POI center, then the value is close to 1. The other information is from the historical data obtained during the user's previous participation in crowdsensing services. In opportunistic crowdsensing services, the service providers need historical data to judge the value of the users. We use $\alpha_{im} \in Z$ to represent the difference between the number of successful and unsuccessful data collection attempts by user i at POI m in the historical data and use the *sigmoid* function $\frac{1}{1+e^{-\alpha_{im}}}$ to represent it. Because the events represented by these two types of information are independent, we multiply the two probabilities to represent the probability that user i can successfully collect data at POI m as p_{im}.

$$p_{im} = \frac{\iint_{(x,y)\in D} e^{-\sqrt{(x-x_m)^2+(y-y_m)^2}} pr_i(x, y)dxdy}{1 + e^{-\alpha_{im}}} \in (0, 1). \tag{1}$$

Therefore, the prior probability model that user i successfully collects data at M POIs is defined by the vector $\boldsymbol{p_i} = (p_{i1}, p_{i2}, \dots, p_{iM})$; meanwhile, the reward that user i expects to receive after performing the data collection task is represented by b_i. In fact, the determination of b_i is not simple, it is related to 1) the energy consumption cost of collecting data, 2) the communication cost of transmitting the data, and 3) the value of the data. In the current mobile network environment, the costs of 1) and 2) are lower, and are determined, but 3 is not certain; it depends on how the service provider uses the data. If the user only uses 1) and 2) as the basis for bidding, then the bid should be very low. If the user also considers the value of her/his data to the service provider, she/he may make a reasonable bid based on her/his own perception. Moreover, the user may improve bids based on the experience of winning or losing in multiple rounds of participation in the crowdsourcing service. Literature [31] gives a reasonable relationship between cost of the user and the bid. Finally, the request submitted by user i is $\theta_i = (\boldsymbol{p}_i, b_i)$. Different from existing research, in our design, users do not need to submit the POIs that they can cover; this coverage information is represented by an a priori probability model.

The purpose of the crowdsensing problem is to maximize the value generated by the selected users, so we must define the value function of the crowdsensing problem first. Although articles [5], [6] use linear-valued functions, there are still some shortcomings for the problem we want to solve. For the service provider, the value generated by each user is a nonlinear monotonically increasing concave function with diminishing marginal utility; that is, the data obtained earlier are more valuable. The same feature is also reflected in federated learning [31]. We also compare the value function proposed in this article with a linear value function in the experimental part. Therefore, for POI m, we define its value function as

$$V^{(m)}(n) = v_m \times \frac{\sum_{k=1}^{n} \frac{1}{k}}{\sum_{k=1}^{r_m} \frac{1}{k}}, n \in \{1, 2, \ldots, r_m\}, \tag{2}$$

where n indicates that n users participated in the coverage of POI m, and all of them successfully collected data. When $n = r_m$, $V^{(m)}(n) = v_m$ in formula (2); that is, the service provider receives all of the value. However, according to our analysis, the user's POI coverage is determined by a probabilistic model, so there is no guarantee that the data can be successfully collected. Therefore, we modified formula (2), and the improved value function is as follows.

$$V^{(m)}(\mathcal{U}) = v_m \times \frac{\sum_{k=1}^{\min\{r_m, |\mathcal{U}|\}} \frac{p_{im}}{k}}{\sum_{k=1}^{r_m} \frac{1}{k}}, i \in \mathcal{U}. \tag{3}$$

$\mathcal{U}$ is an ordered set of users represented using vector $\mathcal{U} = (u_1, u_2, \ldots u_{N'})$, $N' \leq N$, but the set operations still apply to $\mathcal{U}$. Obviously, when r_m is determined, the denominator of formula (3) is a constant value, and the numerator depends on the smaller of r_m and $|\mathcal{U}|$. When $|\mathcal{U}|$ is greater than r_m, the value of $V^{(m)}(\mathcal{U})$ will not change. We define the value function for the mobile crowdsensing problem as

$$V(\mathcal{U}) = \sum_{m \in \mathcal{M}} V^{(m)}(\mathcal{U}). \tag{4}$$

The literature [9] reports that $V(\mathcal{U})$ and $V^m(\mathcal{U})$ are ordered submodularity functions, and the order in set $\mathcal{U}$ affects the result of the function $V(\mathcal{U})$.

Here is a simple example. Suppose that there are 2 POIs, $r_1 = 1$, $r_2 = 2$, and $v_1 = v_2 = 1$, which means that POI 1 must be covered once, POI 2 must be covered twice, and the value obtained by the service provider is 1 in both cases. There are two users u_1 and u_2, and the coverage probabilities for the POIs are $p_{11} = 0.9$, $p_{12} = 0.3$, $p_{21} = 0.3$, and $p_{22} = 0.9$. From formula (3), if we calculate the total value in the order of (u_1, u_2), we obtain

$$V^{(1)}(u_1, u_2) = v_1 \times \frac{\sum_{k=1}^{\min\{r_1, |\mathcal{U}|\}} \frac{p_{i1}}{k}}{\sum_{k=1}^{r_1} \frac{1}{k}} = 0.9,$$

$$V^{(2)}(u_1, u_2) = v_2 \times \frac{\sum_{k=1}^{\min\{r_2, |\mathcal{U}|\}} \frac{p_{i2}}{k}}{\sum_{k=1}^{r_2} \frac{1}{k}} = \frac{0.3 + 0.9/2}{3/2} = 0.5.$$

Thus, $V(u_1, u_2) = V^{(1)}(u_1, u_2) + V^{(2)}(u_1, u_2) = 1.4$. If we calculate the total value in the order of (u_2, u_1), we obtain

$$V^{(1)}(u_2, u_1) = v_1 \times \frac{\sum_{k=1}^{\min\{r_1, |\mathcal{U}|\}} \frac{p_{i1}}{k}}{\sum_{k=1}^{r_1} \frac{1}{k}} = 0.3,$$

$$V^{(2)}(u_2, u_1) = v_2 \times \frac{\sum_{k=1}^{\min\{r_2, |\mathcal{U}|\}} \frac{p_{i2}}{k}}{\sum_{k=1}^{r_2} \frac{1}{k}} = \frac{0.9 + 0.3/2}{3/2} = 0.7.$$

Thus, $V(u_2, u_1) = V^{(1)}(u_2, u_1) + V^{(2)}(u_2, u_1) = 1.0$.

3.2 Ordered Submodularity Properties

The ordered submodularity property is a generalization of the submodularity property, and we must prove that the system model has the properties of both.

Definition 1. Monotone submodular function. *Define $\mathcal{U}$ as a finite set. For any $\mathcal{U}_1 \subseteq \mathcal{U}_2 \subseteq \mathcal{U}$ and $x \in \mathcal{U} \backslash \mathcal{U}_2$, the function $f : 2^{\mathcal{U}} \mapsto R$ is called a monotone submodular function if and only if $f(\mathcal{U}_1 \cup \{x\}) - f(\mathcal{U}_1) \geq f(\mathcal{U}_2 \cup \{x\}) - f(\mathcal{U}_2)$ and $f(\mathcal{U}_1) \leq f(\mathcal{U}_2)$.*

Definition 2. Ordered submodular function. *A sequence function $f(.)$ is ordered submodular if for all sequences A and B, the following property holds for all elements s and $\bar{s}$ [9]:*

$$f(A||s) - f(A) \geq f(A||s||B) - f(A||\bar{s}||B),$$

where $A||s$ represents adding element s after sequence A. Ordered submodular functions can be viewed as a generalization of the monotone submodular function. If we set $s = \bar{s}$, then this means that

$$f(A||s) - f(A) \geq 0,$$

and if we set $\bar{s} = null$, then this means that

$$f(A||s) - f(A) \geq f(A||s||B) - f(A||B),$$

which satisfies the monotone submodular property.

Lemma 1. *$V^m(\mathcal{U})$ is a monotone submodular function.*

Proof. Suppose that $\mathcal{U}_1 \subseteq \mathcal{U}_2 \subseteq \mathcal{U}, i \in \mathcal{U}\backslash\mathcal{U}_2$; we have

$$\begin{aligned} &V^{(m)}(\mathcal{U}_1 \cup \{i\}) - V^{(m)}(\mathcal{U}_1) \\ &= v_m \times \left(\frac{\sum_{k=1}^{\min\{r_m, |\mathcal{U}_1 \cup \{i\}|\}} \frac{p_{im}}{k}}{\sum_{k=1}^{r_m} \frac{1}{k}} - \frac{\sum_{k=1}^{\min\{r_m, |\mathcal{U}_1|\}} \frac{p_{im}}{k}}{\sum_{k=1}^{r_m} \frac{1}{k}} \right) \\ &= \begin{cases} v_m \times \frac{\frac{p_{im}}{|\mathcal{U}_1 \cup \{i\}|}}{\sum_{k=1}^{r_m} \frac{1}{k}}, & r_m - |\mathcal{U}_1| > 0 \\ 0, & otherwise \end{cases}, \end{aligned}$$

and

$$\begin{aligned} &V^{(m)}(\mathcal{U}_2 \cup \{i\}) - V^{(m)}(\mathcal{U}_2) \\ &= v_m \times \left(\frac{\sum_{k=1}^{\min\{r_m, |\mathcal{U}_2 \cup \{i\}|\}} \frac{p_{im}}{k}}{\sum_{k=1}^{r_m} \frac{1}{k}} - \frac{\sum_{k=1}^{\min\{r_m, |\mathcal{U}_2|\}} \frac{p_{im}}{k}}{\sum_{k=1}^{r_m} \frac{1}{k}} \right) \\ &= \begin{cases} v_m \times \frac{\frac{p_{im}}{|\mathcal{U}_2 \cup \{i\}|}}{\sum_{k=1}^{r_m} \frac{1}{k}}, & r_m - |\mathcal{U}_2| > 0 \\ 0, & otherwise \end{cases}, \end{aligned}$$

that is, $V^{(m)}(\mathcal{U}_1 \cup \{i\}) - V^{(m)}(\mathcal{U}_1) \geq V^{(m)}(\mathcal{U}_2 \cup \{i\}) - V^{(m)}(\mathcal{U}_2)$. Furthermore, we have

$$V^{(m)}(\mathcal{U}_1) = \frac{\sum_{k=1}^{\min\{r_m, |\mathcal{U}_1|\}} \frac{p_{im}}{k}}{\sum_{k=1}^{r_m} \frac{1}{k}} \leq \frac{\sum_{k=1}^{\min\{r_m, |\mathcal{U}_2|\}} \frac{p_{im}}{k}}{\sum_{k=1}^{r_m} \frac{1}{k}} = V^{(m)}(\mathcal{U}_2).$$

Thus, $V^{(m)}(\mathcal{U})$ is a monotone submodular function. □

Lemma 2. *$V^m(\mathcal{U})$ is an ordered submodularity function.*

Proof. According to Definition 2, $V_m(\mathcal{U})$ is an ordered submodular function; then, for any sequence A and B in the set $\mathcal{U}$ and any element i and j, we must prove

$$V^{(m)}(A||i) - V^{(m)}(A) \geq V^{(m)}(A||i||B) - V^{(m)}(A||j||B).$$

According to the definition of the $V_m(\mathcal{U})$ function,

$$\begin{aligned} &V^{(m)}(A\|i) - V^{(m)}(A) \\ &= v_m \times \left(\frac{\sum_{k=1}^{\min\{r_m, |A \cup \{i\}|\}} \frac{p_{im}}{k}}{\sum_{k=1}^{r_m} \frac{1}{k}} - \frac{\sum_{k=1}^{\min\{r_m, |A|\}} \frac{p_{im}}{k}}{\sum_{k=1}^{r_m} \frac{1}{k}} \right) \\ &= \begin{cases} v_m \times \frac{\frac{p_{im}}{|A \cup \{i\}|}}{\sum_{k=1}^{r_m} \frac{1}{k}}, & r_m - |A| > 0 \\ 0, & otherwise \end{cases} \end{aligned}$$

$$\begin{aligned} &V^{(m)}(A\|i\|B) - V^{(m)}(A\|j\|B) \\ &= v_m \times \left(\frac{\sum_{k=1}^{\min\{r_m, |A \cup \{i\} \cup B|\}} \frac{p_{im}}{k}}{\sum_{k=1}^{r_m} \frac{1}{k}} - \frac{\sum_{k=1}^{\min\{r_m, |A \cup \{j\} \cup B|\}} \frac{p_{im}}{k}}{\sum_{k=1}^{r_m} \frac{1}{k}} \right) \\ &= \begin{cases} v_m \times \frac{\frac{p_{im} - p_{jm}}{|A \cup \{i\}|}}{\sum_{k=1}^{r_m} \frac{1}{k}}, & r_m - |A| > 0 \\ 0, & otherwise \end{cases}, \end{aligned}$$

that is, $V^{(m)}(A||i) - V^{(m)}(A) \geq V^{(m)}(A||i||B) - V^{(m)}(A||j||B)$; therefore, $V^m(\mathcal{U})$ is an ordered submodular function. □

Theorem 1. *$V(\mathcal{U})$ is an ordered submodular function.*

Proof. According to Definition 2, $V(\mathcal{U})$ is an ordered submodular function; then, for any sequence A and B in the set $\mathcal{U}$ and any element i and j, we must prove

$$V(A||i) - V(A) \geq V(A||i||B) - V(A||j||B).$$

According to Lemma 2 and formula (4), accumulating $V_m(\mathcal{U})$ yields

$$\begin{aligned} &\sum_{m \in \mathcal{M}} V^{(m)}(A||i) - \sum_{m \in \mathcal{M}} V^{(m)}(A) \\ &\geq \sum_{m \in \mathcal{M}} V^{(m)}(A||i||B) - \sum_{m \in \mathcal{M}} V^{(m)}(A||j||B), \end{aligned}$$

that is,

$$V(A||i) - V(A) \geq V(A||i||B) - V(A||j||B).$$

Therefore, $V(\mathcal{U})$ is an ordered submodular function. □

3.3 Preliminaries of the Mechanism Design

Our goal is to find the ordered subset $\mathcal{U} \subseteq \mathcal{N}$ that maximizes the sum of the expected value of each POI under the constraints, which is equivalent to the following model:

$$\text{Maximize} \quad V(\mathcal{U}) = \sum_{m \in \mathcal{M}} V^{(m)}(\mathcal{U}), \quad \mathcal{U} \subseteq \mathcal{N} \tag{5}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{U}} p_i \leq B \tag{5a}$$

$$p_i \geq b_i, \ \forall i \in \mathcal{U}, \tag{5b}$$

(5a) means that the total payment paid to the winners is less than the budget, and (5b) means that the payment for each winning user is greater than her/his bid.

Formula (5) is an ideal problem model. However, in practice, users are selfish and may submit untruthful bids for greater benefits. The value of the mechanism design is based on this characteristic. Specifically, to encourage users to participate in the auction process, the mechanism must satisfy individual rationality. To prevent users from submitting untruthful bids, the mechanism must satisfy truthfulness. To ensure that payments do not exceed the budget, the mechanism must be budget feasible. Additionally, to quickly obtain the allocation and payment solution, the mechanism must satisfy computational efficiency.

We use $\theta_i = (\boldsymbol{p}_i, b_i)$ to denote the true request of user i and $\theta'_i = (\boldsymbol{p}_i, b'_i)$ to denote the declared request of user i. Additionally, we assume that the user may lie about her/his bid so that $b'_i \neq b_i$. We do not discuss the situation where users untruthfully report a prior probability model for POIs $\boldsymbol{p}_i = (p_{i1}, p_{i2}, \ldots, p_{iM})$ because the probability information is not provided by users but collected by the system. Compared with existing studies [5] and [6], this approach reduces the risk of users lying. We use $\boldsymbol{\theta}' = \{\theta'_1, \ldots, \theta'_N\}$ and $\boldsymbol{\theta}'_{-i} = \{\theta'_1 \ldots \theta'_{i-1}, \theta'_{i+1} \ldots, \theta'_N\}$ to denote the declared requests of users submitted to the system and $\boldsymbol{\theta} = \{\theta'_{-i}, \theta_i\}$.

User utility is an important measure to determine the value obtained in an auction, and the user always wants to maximize her/his utility in an auction. User utility is typically expressed as a function. In this article, we assume that user i has the following utility function:

$$u_i(\boldsymbol{\theta}') = \begin{cases} p'_i - b_i, & \text{User } i \text{ is winner in the auction} \\ 0, & otherwise \end{cases}, \quad (6)$$

${p'}_i$ is the final payment paid by the service provider to user i when she/he submits request ${\theta'}_i = (\boldsymbol{p}_i, {b'}_i)$. If the user loses the auction, then the utility is 0. Based on the above description, an individually rational and truthful auction mechanism that is budget feasible can be defined.

Definition 3. Individual rationality. *If a mechanism ensures individual rationality, it should satisfy the condition that when the user submits a truthful request $\theta_i = (\boldsymbol{p}_i, b_i)$, her/his utility will be greater than or equal to zero, i.e., $u_i(\boldsymbol{\theta}) \geq 0$. In other words, as long as the user participates in the auction and reports her/his truthful request, she/he will never incur a loss [32].*

Definition 4. Monotonicity. *If an allocation algorithm ensures monotonicity, it should satisfy that given user request $\theta_i = (\boldsymbol{p}_i, b_i)$, if user i wins by bidding b_i, then she/he will also win by bidding ${\theta'}_i = (\boldsymbol{p}_i, {b'}_i)$ when $b_i \geq {b'}_i$. In a reverse auction, the lower the user's bid is, the easier it is for him/her to win.*

Definition 5. Critical value. *If user i is the winner in crowdsensing, then there must exist a critical value cv_i. If user i's bid $b_i < cv_i$, then user i also wins; otherwise, user i loses.*

Definition 6. Truthfulness (Myerson's theorem). *If a mechanism is truthful, it implies that for every user i, given a truthful declaration request θ_i and declaration requests $\boldsymbol{\theta'}_{-i}$ of the other users, we can obtain $u_i(\boldsymbol{\theta'}_{-i}, \theta_i) \geq u_i(\boldsymbol{\theta'}_{-i}, \theta'_i)$, which is equivalent to $u_i(\boldsymbol{\theta}) \geq u_i({\boldsymbol{\theta'}}_i)$. Therefore, submitting a truthful request is the dominant strategy for each user [33]. The previous literature [34] notes that if the allocation function of a mechanism satisfies monotonicity and the payment function satisfies critical value theory, then the mechanism is truthful.*

Definition 7.
Budget feasibility. *If an auction mechanism is budget feasible, then the sum of the payments of all the winners must not exceed the budget limit B proposed by the service provider; that is, $\sum_{i \in \mathcal{U}} p_i \leq B$[10].*

Definition 8. Computational efficiency. *If an algorithm is computationally efficient, it can be executed in polynomial time. Because obtaining optimal solutions to general submodular problems may take exponential time.*

4 Ordered Submodularity-Proportional Share Mechanism (O-PSM)

In this section, we first introduce the design of the algorithm, then analyze and give examples of the algorithm, and finally prove that the algorithm conforms to the economic characteristics of the mechanism design. At the same time, we give the frequently used notations in Table 1.

4.1 Design of O-PSM

Inspired by the PSM [10] algorithm, we designed the O-PSM algorithm. O-PSM can solve the winner decision (also known as allocation) problem and payment price calculation with ordered-submodularity features while satisfying individual rationality, truthfulness, budget feasibility and computational efficiency, which we will prove later. The basic idea of the O-PSM algorithm is to select the user with the highest marginal value density and add it to the current user sequence at each step in the allocation stage (to ensure the monotonicity of the allocation) until the user's bid exceeds the budget threshold (to ensure budget feasibility). In the payment phase, the maximum bid that allows the user to still win is the payment the service provider pays the user (guaranteed critical value theory). The main

TABLE 1
Frequently Used Notations

Notation	Implication
$\mathcal{U}$	User set.
$\mathcal{M}$	POI set.
D	Areas where service providers need to collect data.
p_{im}	Probability that user i successfully collects data at POI m.
b_i	Bid of user i.
α_{im}	Difference between the number of successful and unsuccessful data collection attempts by user i at POI m in the historical data.
$pr_i(x,y)$	Probability of user i appearing at point $(x,y) \in D$.
A	User sequence generated in the allocation stage of the algorithm.
A'	User sequence generated in the payment stage of the algorithm.
A_i	In the allocation stage, the user sequence composed of the first i users.
${A'}_i$	In the payment stage, the user sequence composed of the first i users.
$A_{i\text{-}1} \Vert i$	In the allocation stage, the sequence of the first $i-1$ users followed by user i.
${A'}_{i\text{-}1} \Vert i$	In the payment stage, the sequence of the first $i-1$ users followed by user i.
$V(A)$	Total value generated by user sequence A.
$V_i(A),\ V_i$	$= V(A_{i-1} \Vert i) - V(A_{i-1})$, In user sequence A, the marginal value generated by user i.
$V_{i\vert A}$	In the allocation stage, adding user i following the user sequence A, the marginal value generated by user i.
$V_{i\vert A'}$	In the payment stage, adding user i follows user sequence A, the marginal value generated by user i.
$V_{i(j)}$	= $V_{i\vert {A'}_{j-1}}$. When calculating the payment for i in the payment stage, if user i replaces user j, the marginal value generated by user i.
$V_{i_j\vert A'_{j-1}}$	= $V_{i_j\vert {A'}_{j-1}}$. When calculating the payment for i in payment stage, the marginal value generated by user j.
p_i	Payment the service provider finally pays to user i.

difference between the O-PSM and PSM algorithms is that in PSM, the value function satisfies only submodularity, while in O-PSM, the value function also satisfies ordered-submodularity and can solve the problems proposed in the article. Algorithm 1 is the O-PSM algorithm.

Algorithm 1. Ordered Submodularity-Proportional Share Mechanism (O-PSM)

Input: $\mathcal{N}, \boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_N), B$
Output: $A, \boldsymbol{p}$
1: $A \leftarrow \phi, i \leftarrow \arg\max_{j\in\mathcal{U}}(\frac{V_{j|A}}{b_j}), \boldsymbol{p} \leftarrow (p_1, p_2, \ldots, p_N)$
2: **while** $b_i \le \frac{B}{2} \times \frac{V_{i|A}}{V(A||i)}$ **do**
3: $A \leftarrow A||i, i \leftarrow \arg\max_{j\in\mathcal{N}\setminus A}\left(\frac{V_{j|A}}{b_j}\right)$
4: **end while**
5: **for** each $i \in \mathcal{N}$ **do**
6: $p_i \leftarrow 0$
7: **end for**
8: **for** each $i \in A$ **do**
9: $\mathcal{N}' \leftarrow \mathcal{N}\setminus\{i\}$, $A' \leftarrow \phi$
10: **repeat**
11: $i_j \leftarrow \arg\max_{j\in\mathcal{N}'\setminus A'}\left(\frac{V_{j|A'}}{b_j}\right)$
12: $p_i \leftarrow \max\{p_i, \min\{b_{i(j)}, \rho_{i(j)}\}\}$
13: $A' \leftarrow A'||i_j$
14: **until** $b_{i_j} \le \frac{B}{2} \times \frac{(V(A')-V(A'\setminus i_j))}{V(A')}$
15: **if** i is the last winner in A **then**
16: $p_i \leftarrow b_i$
17: **end if**
18: **end for**
19: **return** $A, \boldsymbol{p}$

In Algorithm 1, lines 1-4 are the allocation stage, and lines 5-18 are the payment stage. In the allocation stage, we define $V_{i|A} = V(A||i) - V(A)$ as the marginal value of user i after being added to sequence A. For simplicity, we use V_i instead of $V_{i|A}$ when there is no ambiguity. Define $\frac{V_i}{b_i}$ as the marginal value density of user i. To satisfy monotonicity, we must sort the marginal value densities according to the following rules. Initially, $A \leftarrow \phi$, and the first user is $i \in \arg\max_{j\in\mathcal{U}}(V_{j|\phi}/b_j)$, assuming that the first i users are all determined, that is, $A = (1, 2, \ldots, i)$. The determination method of the i+1-th user is $\arg\max_{j\in\mathcal{U}\setminus A}(V_{j|A}/b_j)$, so we can obtain a sequence in nonincreasing order of the user's marginal value density

$$\frac{V_1}{b_1} \ge \frac{V_2}{b_2} \ge \ldots \ge \frac{V_N}{b_N}. \tag{7}$$

The algorithm selects the winning user under the premise of satisfying $b_i \le \frac{B}{2}\frac{V_{i|A}}{V(A||i)}$ and generates the final winning user sequence A.

In the payment stage, we calculate the payment for each winning user. First, the winning user $i \in A$ is removed from the user set $\mathcal{U}$; then, the winner decision is run for the remaining users to generate a winning user sequence, A', that does not include user i. The payment for user i is calculated based on A'. $\min\{b_{i(j)}, \rho_{i(j)}\}$ represents the maximum bid (according to the allocation rules, if the bid of user i is higher than $\min\{b_{i(j)}, \rho_{i(j)}\}$, then she/he cannot win at the j-th position) that user i needs to appear and win at the j-th position during the payment stage, where

$$\begin{aligned} V_{i(j)} &= V(A'_{j-1}||i) - V(A'_{j-1}) \\ V_{i_j|A'_{j-1}} &= V(A'_{j-1}||i_j) - V(A'_{j-1}) \\ b_{i(j)} &= \frac{V_{i(j)} \times b_{i_j}}{V_{i_j|A'_{j-1}}} \\ \rho_{i(j)} &= \frac{V_{i(j)}}{V(A'_{j-1}||i)} \times \frac{B}{2}. \end{aligned} \tag{8}$$

In formula (8), $V_{i(j)}$ means, when calculating the payment for i in the payment stage, the marginal value generated by user i if user i replaces user j. $V_{i_j|A'_{j-1}}$ means, when calculating the payment for i in the payment stage, the marginal value generated by user j. We also call $\rho_{i(j)}$ the threshold value. As j increases, $\rho_{i(j)}$ decreases, but as j increases, $V_{i(j)}$ decreases and $\frac{b_{i_j}}{V_{i_j|A'_{j-1}}}$ increases; thus, the change trend of $b_{i(j)}$ cannot be determined. Therefore, to ensure the truthfulness of the mechanism, we take the maximum value of user i's bids at all the winning positions as the final payment paid to user i. Suppose that k' is the last position that satisfies $b_j \le \frac{(V(A')-V(A'\setminus j))}{V(A')} \times \frac{B}{2}$, $j \in \mathcal{U}\setminus\{i\}$ in the payment stage, and define $r \in \{1, 2, \ldots, k'+1\}$ as the position where p_i reaches the maximum value. At this time, $p_i = \min\{b_{i(r)}, \rho_{i(r)}\}$, so user i can be paid the maximum payment price when she/he is selected, and user i has no incentive to lie about her/his bid. We will prove these points later. In practice, service providers can pay as soon as the algorithm is executed or provide a payment according to the actual number of tasks completed by the user when the crowdsensing service ends. The payment calculated in Algorithm 1 can be used as the basis for the final payment.

4.2 Analysis and Example of O-PSM

Most mechanism designs include an allocation (winner decision) stage and a payment stage and need to meet the economic characteristics of individual rationality and truthfulness. For the PSM algorithm in [10], the most valuable users are selected by a greedy strategy in the allocation stage, and in the payment stage, to ensure the truthfulness of the user's bid, the highest profit that the user may obtain is used as the payment price. The core design idea can be traced back to the monotonic allocation and critical value theory in the literature [35]. Similar designs are widely used in mechanism design articles in different fields, such as [16], [31], [32]. Notably, the largest difference between such algorithms is the decision basis for the monotonic allocation. For example, in [16], the dominant resource ratio is used as the decision basis, while in [32], the resource density is used as the decision basis. This reflects the algorithm designer's understanding of the optimization problem. In the PSM algorithm, the decision basis is the marginal utility of the submodularity function divided by the user bid, that is, $\frac{V(\mathcal{U}\cup\{i\})-V(\mathcal{U})}{b_i}$. However, in the O-PSM algorithm, the decision basis is the marginal utility of the ordered submodularity function divided by the user bid, that is, $\frac{V(A||i)-V(A)}{b_i}$. Although the two equations look very similar, there are essential differences in the decision-making principle because the order of users in the set $\mathcal{U}$ in the PSM algorithm does not affect the value of $V(\mathcal{U})$, while in O-PSM, even though sequences A and A' have the same users,

TABLE 2
Example of O-PSM

Allocation stage	$v_{1\|A}, v_{2\|A}, v_{3\|A}$	$\frac{v_{1\|A}}{b_1}, \frac{v_{2\|A}}{b_2}, \frac{v_{3\|A}}{b_3}$	A	V
Round1	$(2.9, 4.4, 4.2)$	$(0.97, 2.2, 1.4)$	u_2	4.4
Round2	$(1.0, --, 1.2)$	$(0.33, --, 0.4)$	u_2, u_3	5.6
Payment stage	A'	$b_{i(j)}$	$\rho_{i(j)}$	$\min\{b_{i(j)}, \rho_{i(j)}\}$
Calculate the price paid to u_2	u_3	3.14	20	3.14
	u_3, u_1	3.0	3.85	3.0
			$p_2 = \max\{3.14, 3.0\} = 3.14$	
Calculate the price paid to u_3	u_2	1.91	20	1.91
	u_2, u_1	3.6	4.29	3.6
			$p_3 = \max\{1.91, 3.6\} = 3.6$	

the values of $V(A)$ and $V(A')$ are not equal, which is described in formula (7) and affects the calculation process of the payment stage of the O-PSM algorithm (lines 8-18 of Algorithm 1).

Consider a simple example. Suppose that there are 2 POIs, m_1 and m_2. m_1 needs to be covered once, and the expected value is 3, denoted as $m_1 = (1, 3)$. Similarly, $m_2 = (2, 6)$. There are 3 users u_1, u_2 and u_3, and the request submitted by user 1 is $\theta_1 = (\boldsymbol{p}_1, b_1) = ((0.3, 0, 5), 3)$, which means that the probability of user 1 covering m_1, m_2 is (0.3,0,5), and the expected reward is 3. Similarly, $\theta_2 = ((0.8, 0, 5), 2)$, and $\theta_3 = ((0.6, 0, 6), 3)$. The service provider's budget is 40. The allocation and payment process is shown in Table 2.

In the allocation phase, the marginal value densities of the three users are first calculated, and the winning user sequence is ϕ. The marginal value of user 1 is $\frac{0.3}{1} \times 3 + \frac{0.5}{1.5} \times 6 = 2.9$, and the marginal value density is $\frac{2.9}{3} = 0.97$. Similarly, the marginal value of user 2 is $\frac{0.8}{1} \times 3 + \frac{0.5}{1.5} \times 6 = 4.4$, the marginal value density is $\frac{4.4}{2} = 2.2$, the marginal value of user 3 is $\frac{0.6}{1} \times 3 + \frac{0.6}{1.5} \times 6 = 4.2$, and the marginal value density is $\frac{4.2}{3} = 1.4$. Therefore, the winning user in round 1 is u_2. The algorithm then enters round 2. Because u_2 has already won, its marginal value and marginal value density do not need to be calculated in round 2. In round 2, the marginal values of u_1 and u_3 are $(1.0, 1.2)$, and the marginal value densities are $(0.33, 0.4)$. u_3 has a higher marginal value density and thus wins round 2. Because the maximum number of POI coverages is 2, regardless of whether a user is added to the current sequence, the total value cannot be increased; thus, the allocation process is finished, and the winning user sequence is $A = (u_2, u_3)$.

In the payment stage, the algorithm first calculates the price paid to u_2. After removing u_2, the marginal value density of u_3 is the largest, so $A' = (u_3)$. At this time, if u_2 occupies the position of u_3, then her/his bid is at most 3.14. Meanwhile, the threshold value is 20. The minimum of the two is 3.14, so u_2 bids at most 3.14 in this round. Then, based on the current user sequence $A' = (u_3)$, the user with the highest marginal value density is still u_1; therefore, $A' = (u_3, u_1)$. According to the algorithm, if u_2 occupies the position of u_1, then the bid is at most 3, and the threshold value is 3.85. The minimum of the two is 3.0, so u_2 bids at most 3 in this round. Finally, the price paid to u_2 is $p_2 = \max\{3.14, 3.0\} = 3.14 > b_2 = 2$. Similarly, the price that can be paid to u_3 is $p_3 = \max\{1.9, 3.6\} = 3.6 > b_3 = 3$. Notably, if we set u_2's bid to greater than 3.14 or set u_3's bid to greater than 3.6, they cannot win, which reflects the truthfulness of the mechanism, as proven below.

4.3 Properties of O-PSM

Lemma 3. *O-PSM satisfies monotonicity.*

Proof. The winner decision is made according to the nonincreasing order of the marginal value density. When user i bids $b'_i < b_i$, $\frac{V_i}{b'_i} > \frac{V_i}{b_i}$ is satisfied, and $\frac{V_i}{V(A||i)} \times \frac{B}{2}$ monotonically decreases, satisfying $b'_i < b_i \leq \frac{V_i}{V(A||i)} \times \frac{B}{2}$. Thus, user i must also win, and O-PSM satisfies monotonicity. □

Lemma 4. *O-PSM satisfies critical value theory.*

Proof. To show that O-PSM satisfies critical value theory, it needs to be proven that if user i wins and changes her/his bid such that $b'_i < p_i$, then user i still wins, whereas if $b'_i > p_i$, then user i loses. According to Algorithm 1, the price paid to user i can be equivalently defined as

$$p_i = \max_{j \in \{1, 2, \ldots, k'+1\}} \{\min\{b_{i(j)}, \rho_{i(j)}\}\}, \tag{9}$$

where k' is the last position that satisfies $b_j \leq \frac{(V(A') - V(A' \backslash j))}{V(A')} \times \frac{B}{2},\ j \in \mathcal{U} \backslash \{i\}$ in the payment stage. Define $r \in \{1, 2, \ldots, k'+1\}$ as the position where p_i obtains the maximum value, that is, $p_i = \min\{b_{i(r)}, \rho_{i(r)}\}$.

1) If user i bids $b'_i < p_i = \min\{b_{i(r)}, \rho_{i(r)}\}$, because $b'_i < b_{i(r)}$, user i will be selected earlier than or at the r-th position. Meanwhile, because $b'_i < \rho_{i(r)}$, user i will still win.
2) If user i bids $b'_i > p_i = \min\{b_{i(r)}, \rho_{i(r)}\}$, then there are two cases:
   - Suppose that $b_{i(r)} \leq \rho_{i(r)}$, which means $\min\{b_{i(r)}, \rho_{i(r)}\} = b_{i(r)}$. In the payment stage, a higher bid will cause user i to be processed after r; if $b_{i(r)} = \max_{j \in \{1, 2, \ldots, k'+1\}} \{b_{i(j)}\}$ at this time, then $b'_i > b_{i(r)}$ will cause i to lose (user i has no value to the service provider). Otherwise, if $b_{i(r)} < b_{i(j)},\ j \in \{1, 2, \ldots, k'+1\}$, then there must be $\rho_{i(j)} < b_{i(r)} < b_{i(j)}$ (otherwise, p_i is not equal to $\min\{b_{i(r)}, \rho_{i(r)}\}$). At this time, user i's bid b'_i exceeds all the threshold values $\rho_{i(j)}, \forall j \in \{1, 2, \ldots, k'+1\}$, and user i cannot win.
   - Suppose that $\rho_{i(r)} < b_{i(r)}$, which means $\min\{b_{i(r)}, \rho_{i(r)}\} = \rho_{i(r)}$. If $\rho_{i(r)} = \max_{j \in \{1, 2, \ldots, k'+1\}} \{\rho_{i(j)}\}$, then $b'_i > \rho_{i(r)}$ will cause i to lose (user i's bid has exceeded the maximum threshold). Otherwise, if $\rho_{i(r)} < \rho_{i(j)},\ j \in$

$\{1, 2, \ldots, k' + 1\}$, then there must be $b_{i(j)} < \rho_{i(r)} < \rho_{i(j)}$. At this time, $b'_i > \rho_{i(r)} > b_{i(j)} \Rightarrow b'_i > b_{i(j)}$ will cause user i to be placed after the j-th position $\forall j \in \{1, 2, \ldots, k' + 1\}$, which means that the mechanism will not consider user i in all positions, so user i cannot win.

In summary, if user i's bid $b'_i < p_i = \min\{b_{i(r)}, \rho_{i(r)}\}$, then she/he can win, and if $b'_i > p_i = \min\{b_{i(r)}, \rho_{i(r)}\}$, then she/he cannot win; thus, O-PSM satisfies critical value theory. □

Theorem 2. *O-PSM is truthful.*

Proof. According to Definition 6, Lemmas 3 and 4, O-PSM is truthful. □

Theorem 3. *O-PSM satisfies individual rationality.*

Proof. If the mechanism satisfies individual rationality, then we must prove $b_i \le p_i$. Considering the winning user i in the allocation stage, when calculating the payment of i in the set $\mathcal{U}' \leftarrow \mathcal{U}\backslash\{i\}$, the position of i is replaced by user j. According to the algorithm, the user sequence before i in the allocation stage is consistent with the user sequence before j in the payment stage, that is, $A_{i-1} = A'_{j-1}$; therefore,

$$\rho_{i(j)} = \frac{V_{i|A'_{j-1}}}{V(A'_{j-1}||i)} \times \frac{B}{2} = \frac{V_{i|A_{i-1}}}{V(A_{i-1}||i)} \times \frac{B}{2} \ge b_i.$$

Furthermore, in the allocation stage, $b_i \le \frac{V_{i|A_{i-1}} b_j}{V_{j|A_{i-1}}}$, and we can obtain

$$b_{i(j)} = \frac{V_{i|A'_{j-1}} b_j}{V_{j|A'_{j-1}}} = \frac{V_{i|A_{i-1}} b_j}{V_{j|A_{i-1}}} \ge b_i.$$

Thus,

$$b_i \le \min\{b_{i(j)}, \rho_{i(j)}\} \le \max_{j \in \{1,2,\ldots,k'+1\}} \{\min\{b_{i(j)}, \rho_{i(j)}\}\} = p_i.$$

Inspired by [6], we prove that O-PSM is budget feasible. □

Lemma 5. *For the sequence set $A \subset B \subset \mathcal{U}$ composed of users and user $i^* \leftarrow \arg\max_{j \in B\backslash A}(\frac{V_{j|A}}{b_j})$, we obtain*

$$\frac{V(B) - V(A)}{\sum_{i \in B} b_i - \sum_{j \in A} b_j} \le \frac{V_{i*|A}}{b_{i*}}.$$

Proof. We use contradiction; assume that any $i \in B\backslash A$ satisfies

$$\frac{V(B) - V(A)}{\sum_{i \in B} b_i - \sum_{j \in A} b_j} > \frac{V_{i|A}}{b_i}.$$

Considering the subforms for $i \in B\backslash A$, we obtain

$$\frac{V(B) - V(A)}{\sum_{i \in B} b_i - \sum_{j \in A} b_j} > \frac{\sum_{i \in B\backslash A} V_{i|A}}{\sum_{i \in B\backslash A} b_i} = \frac{\sum_{i \in B\backslash A} V_{i|A}}{\sum_{i \in B} b_i - \sum_{j \in A} b_j}.$$

Thus, $V(B) - V(A) > \sum_{i \in B\backslash A} V_{i|A}$, which does not satisfy the properties of submodular functions because

$$V(B) - V(A) = \sum_{i \in B\backslash A, k=0,1,\ldots|B|-|A|-1} V_{i|A+k},$$

and $V_{i|A} \ge V_{i|A+k}$. □

Theorem 4. *O-PSM is budget feasible.*

Proof. If O-PSM is budget feasible, then $\sum_{i \in A_k} p_i \le B$, where A_k represents the sequence sets that win in the allocation stage. Defining $r \in \{1, 2, \ldots, k' + 1\}$ as the position where p_i reaches the maximum value in the payment stage, which must satisfy $p_i = p_{i(r)} = \min\{b_{i(r)}, \rho_{i(r)}\}$, we can obtain

$$p_i = p_{i(r)} \le b_{i(r)} = \frac{V_{i|A'_{r-1}} \times b_r}{V_{r|A'_{r-1}}} \tag{10}$$

$$p_i = p_{i(r)} \le \rho_{i(r)} = \frac{V_{i|A'_{r-1}}}{V(A'_{r-1}||i)} \times \frac{B}{2}. \tag{11}$$

According to Theorem 3 (individual rationality), $b_i \le p_i$. In the allocation stage, user i wins in the i-th position, $b_i \ge p_{i(j)} = \min\{b_{i(j)}, \rho_{i(j)}\}$, $\forall j = 1, 2, \ldots, i-1$. Therefore, $p_{i(r)} \ge p_{i(j)}, \forall j = 1, 2, \ldots, i-1$. In this case, r is at least i and $A_{i-1} \subseteq A'_{r-1}$, so $V_{i|A_{i-1}} \ge V_{i|A'_{r-1}}$ can be obtained. Suppose that $\{A'_{r-1}||i\} \subseteq A_k$.

a) If $\{A'_{r-1}||i\} = A_k$, then according to formula (11),

$$\frac{V_{i|A_{i-1}}}{p_i} \ge \frac{V_{i|A'_{r-1}}}{p_i} \ge V(A'_{r-1}||i) \times \frac{2}{B} = V(A_k) \times \frac{2}{B} \ge \frac{V(A_k)}{B}.$$

We can obtain $p_i \le \frac{V_{i|A_{i-1}}}{V(A_k)} \times B = \frac{V_i}{V(A_k)} \times B$, that is, $p_i \le \frac{V_i}{V(A_k)} B,\ i \in A_k$

b) For $\{A'_{r-1}||i\} \subset A_k$, we use proof by contradiction, assuming $p_i > \frac{V_i}{V(A_k)} B,\ i \in A_k$. Suppose that

$$r^* = \arg\max_{j \in A_k \backslash \{A'_{r-1}||i\}} \left(\frac{V_{j|(A'_{r-1}||i)}}{b_j}\right).$$

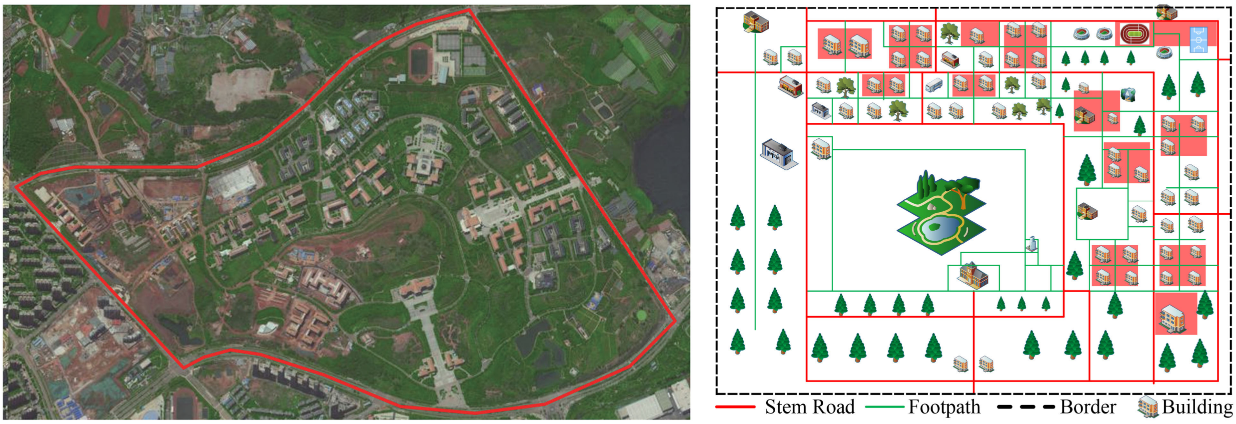
We can obtain

$$\frac{V_{r^*|(A'_{r-1}||i)}}{b_{r^*}} \le \frac{V_{r^*|(A'_{r-1})}}{b_{r^*}} \le \frac{V_{r|(A'_{r-1})}}{b_r}. \tag{12}$$

The first two terms of the inequality are due to the diminishing marginal value, and the last two terms are due to user r being the user with the highest marginal value density in the r-th position. Thus, according to formulas (10) and (12) and Lemma 5,

$$\frac{V(A_k) - V(A'_{r-1}||i)}{\sum_{i \in A_k} b_i - \sum_{j \in A'_{r-1}||i} b_j} \le \frac{V_{r*|(A'_{r-1}||i)}}{b_{r*}} \le \frac{V_{r^*|(A'_{r-1})}}{b_{r^*}} \le \frac{V_{r|(A'_{r-1})}}{b_r} \le \frac{V_{i|(A'_{r-1})}}{p_i}. \tag{13}$$

Because we are using contradiction, we assume $p_i > \frac{V_i}{V(A_k)} \times B,\ i \in A_k$. Thus,



(a) Real map (b) Virtual map

Fig. 2. Real map and virtual map of the POI coverage area.

$$\frac{V_{i|(A'_{r-1})}}{p_i} \leq \frac{V_{i|(A'_{r-1})} \times V(A_k)}{V_{i|A_{i-1}} \times B} \leq \frac{V(A_k)}{B}. \quad (14)$$

In the allocation stage, according to $b_k \leq \frac{B}{2} \times \frac{V_k}{V(A_k)}$ and formula (7), we can obtain

$$\frac{V_1}{b_1} \geq \frac{V_2}{b_2} \geq \ldots \geq \frac{V_k}{b_k} \geq \frac{2V(A_k)}{B}.$$

Thus,

$$\sum_{j \in 1,2,\ldots k} b_j \leq \frac{B}{2} \times \frac{\sum_{j \in 1,2,\ldots k} V_j}{V(A_k)} = \frac{B}{2}.$$

We also know that

$$\sum_{i \in A_k} b_i - \sum_{j \in A'_{r-1}||i} b_j = \sum_{j \in A_k \setminus \{A'_{r-1}||i\}} b_j \leq \sum_{j \in A_k} b_j \leq \frac{B}{2}. \quad (15)$$

According to formulas (13), (14) and (15), we can obtain

$$\frac{2(V(A_k) - V(A'_{r-1}||i))}{B} \leq \frac{V(A_k) - V(A'_{r-1}||i)}{\sum_{i \in A_k} b_i - \sum_{j \in A'_{r-1}||i} b_j} \leq \frac{V_{i|(A'_{r-1})}}{p_i} \leq \frac{V(A_k)}{B} \Rightarrow V(A_k) \leq 2V(A'_{r-1}||i).$$

According to formula (11) and $A_{i-1} \subseteq A'_{r-1}$, $V_{i|A_{i-1}} \geq V_{i|A'_{r-1}}$

$$\frac{V_{i|A_{i-1}}}{p_i} \geq \frac{V_{i|A'_{r-1}}}{p_i} \geq V(A'_{r-1}||i) \times \frac{2}{B} \geq \frac{V(A_k)}{B},$$

which contradicts the assumption. Thus,

$$p_i \leq \frac{V_i}{V(A_k)} B, \ i \in A_k.$$

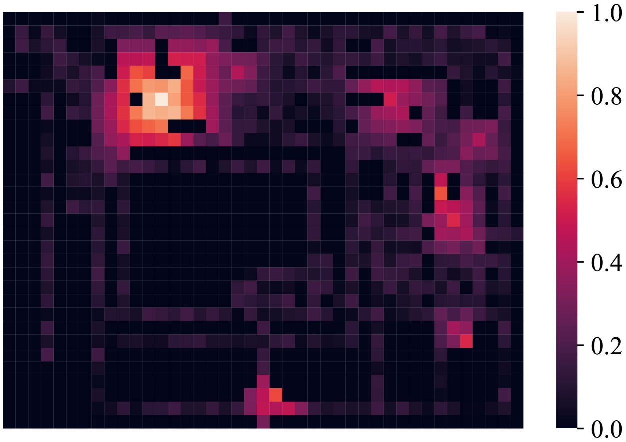
In summary, the upper bound of p_i is $\frac{V_i}{V(A_k)} \times B$. In conclusion, $\sum_{i \in A_k} p_i \leq \sum_{i \in A_k} \frac{V_i}{V(A_k)} \times B = B$; thus, O-PSM is budget feasible. □

Theorem 5. *O-PSM satisfies computational efficiency.*

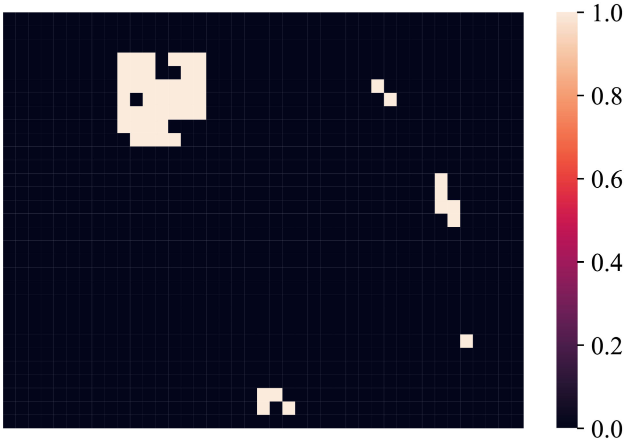
Proof. In the allocation stage, the marginal value density of each user must be sorted. Therefore, the computational complexity is $O(N^2)$. In the payment stage, the payment paid to each winning user is equivalent to calling the allocation stage multiple times; additionally, the time complexity is $O(N^3)$, so O-PSM satisfies computational efficiency. □

5 Numerical Results

This article compares the O-PSM algorithm with the BEACON algorithm [6] and the random algorithm. The BEACON algorithm is a good PSM implementation that has the advantages of a high operating efficiency and easy implementation. We use the BEACON algorithm as our baseline. Although BEACON was originally used to solve the problem of participatory mobile crowdsensing service, the main difference between opportunistic and participatory mobile crowdsensing service is how to determine the user's request. For the opportunistic crowdsensing service, the user's coverage probability of POIs is based on the user information collected by the service provider (see formula (1)), so the credibility is very high. For the participatory crowdsensing service, this information is actively reported by users (the credibility is affected by the user's subjective consciousness). But entering the algorithm part, there is not much difference between the winners' decision and payment calculation. The Beacon algorithm compared in this article has been improved and can be used in the opportunistic crowdsensing service. We conduct six experiments on the user scale, POI scale, budget scale, value function, ε on BEACON and mechanism truthfulness. The analysis and comparison are carried out in multiple dimensions, such as total value, payment, and number of winning users. The results show that the O-PSM algorithm outperforms the existing similar algorithms.



(a) Coverage probability heatmap for O-PSM



(b) Coverage heatmap for BEACON, $\varepsilon = 0.45$

Fig. 3. Coverage probability heatmap.

5.1 Experimental Settings

1) Selection of POIs. Based on the campus map (approximately $3km^2$) shown in Fig. 2a (the campus is located within the red border), the virtual map (Fig. 2b) is created. In the reachable area on the map, a circle with a diameter of 50 meters is used as a POI, and a total of 641 POIs are obtained. For the main roads and main buildings on the campus (canteens, classrooms, dormitories; the red part in Fig. 2b), more instances of coverage are assumed to be required to obtain valid data; that is, the POIs in these locations have larger r_m and v_m values. Therefore, we set $w_m \in \{1, 2, \ldots, 10\}$ for each POI to indicate the importance of its coverage area. The number of coverage times of a POI is $r_m = \lceil 10 \times (w_m \times \sigma) \rceil$, and the generating value is $v_m = \lceil r_m \times \sigma \rceil$, where σ is a random value between 1 and 2.
2) We obtained the main activity area and activity level of 500 students through questionnaires. In the results, the user's main activity area is composed of multiple independent areas, and these areas usually radiate around a building or other location as the center. Based on the above rules, we generate a prior probability model of user coverage of POIs. First, according to this real information, 5,000 user samples are generated by adding random noise. Then, in each sample, a larger coverage probability is assigned according to the user's active center, and this probability linearly decreases with distance. The rate of decrease is determined by the user's activity level. Fig. 3a shows a heatmap of a user's coverage probability of POIs. Combined with Fig. 2b, the main activity area of the user can be seen. Finally, according to the center position of the 641 POIs and formula (1), we calculate the coverage prior probability model of each user for the 641 POIs.
3) We improve the BEACON algorithm since the value function of the BEACON algorithm cannot identify the probability, so we set a threshold ε. Additionally, in the user POI coverage probability model, probabilities greater than the threshold are set to 1, and probabilities less than the threshold are set to 0; that is,

$$p_{im}^{beacon} = \begin{cases} 1, & p_{im} \geq \varepsilon \\ 0, & otherwise \end{cases}.$$

Notably, when $p_{im}^{beacon} = \{0, 1\}$, the value function loses the ordered feature and retains only monotonic submodularity. In the value function of the BEACON algorithm, we still use 0 or 1 to calculate the user's marginal value. In the experiment, we take $\varepsilon = 0.45$ because an excessively high ε will lead to an insufficient number of POIs that the users can cover, which will eventually lead to a low total value of the BEACON algorithm. On the other hand, when ε is too low, it does not conform to the real situation. Fig. 3b shows the heatmap used by the BEACON algorithm, which is completely applied to the coverage model in [6].
4) We use the random algorithm as another comparison algorithm because it is also a strategy for solving the crowdsensing problem and has good stability. In the random algorithm design, in the allocation stage, we do not rank the marginal value or marginal value density of the users but randomly select users from the set of all users to form a sequence and end the algorithm when no user can satisfy the $b_i \leq (B/2)/(V_{i|A}/V(A\|I))$ condition or the payment exceeds the budget. Notably, the random algorithm cannot guarantee truthfulness and individual rationality; therefore, the payment of the random algorithm to the user is the user's bid.
5) All algorithms use the same data, and to eliminate the influence of the randomness of the data, we test each indicator in the experiment 50 times; the average value is shown in the figure.
6) We implement the O-PSM, BEACON, and random algorithms using Python. The hardware configuration of the experimental platform is as follows: the processor is an Intel(R) Core(TM) i5-10210 CPU with 16 GB memory and a 512 GB SSD.

5.2 Experimental Results

5.2.1 *Impact of the User Scale*

In this experiment, the number of users starts at 200 and gradually increases to 2,000 in steps of 200, the service

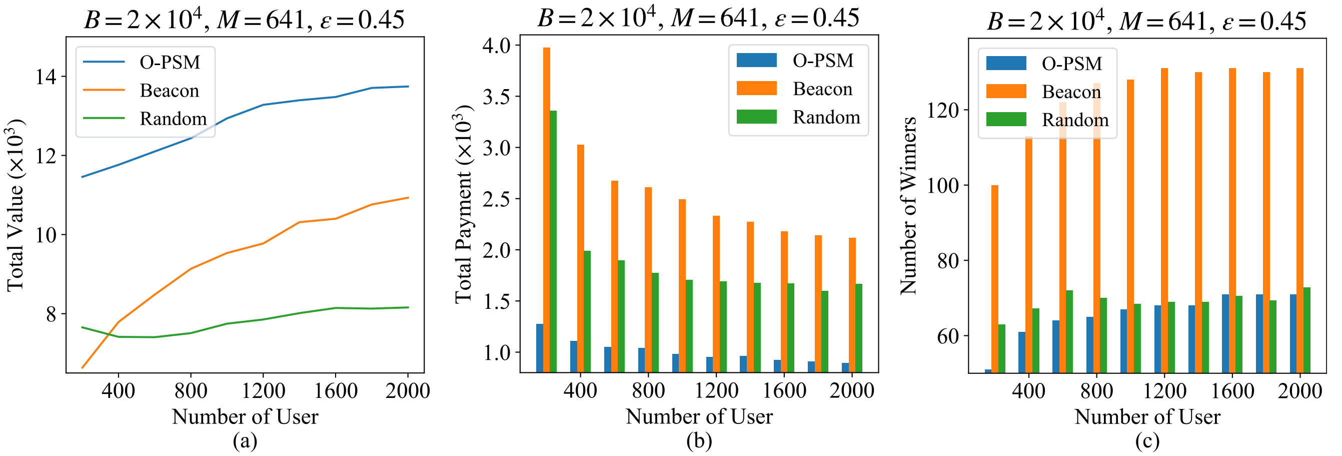


Fig. 4. Impact of the user scale.

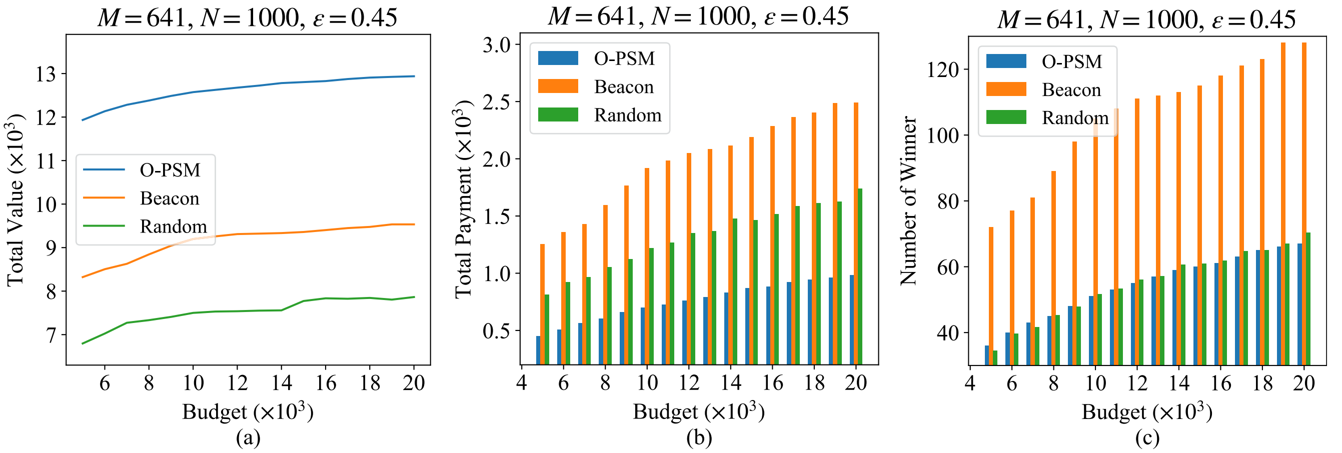


Fig. 5. Impact of the budget.

provider budget is set to 20,000, and the number of POIs is 641. The experimental results are shown in Fig. 4.

The calculation of the total value in Fig. 4a can be obtained based on formulas (3) and (4) and points 1) and 2) of Section 5.1. The value curves of O-PSM and BEACON have diminishing marginal values, but the total value of the O-PSM algorithm is higher because O-PSM is ordered-submodular and can consider the contributions of a large number of the small probability values of the users (long tail effect) in the total value calculation. Although the BEACON algorithm regards user coverage probabilities for POIs greater than 0.45 as 1, compared with the entire campus area, a user's active area is small, which leads to a small number of high-coverage points and ultimately a low total value. This also illustrates the shortcoming of the existing monotonic submodular algorithm in the coverage model based on prior probability. Because the random algorithm selects users randomly, whether new users can be added is mainly determined by $\sum_{i\in\mathcal{U}} p_i \leq B$, so the selection of cost-effective users cannot be guaranteed. In multiple independent repeated experiments, the output value is relatively stable, but the total value is the lowest among the three algorithms.

Fig. 4b shows the payment costs of the three algorithms, and the O-PSM algorithm has the lowest payment because it benefits from ordered-submodularity, which is reasonable when calculating the value of $p_i = \max\{p_i, \min\{b_{i(j)}, \rho_{i(j)}\}\}$. The BEACON algorithm obtains a higher payment price because the coverage probabilities greater than the threshold are set to 1. The random algorithm, by randomly selecting users, leads to a relatively stable payment price after 400. From the perspective of service providers, being able to obtain a higher value under a lower payment is a very good result, which is also one of the advantages of the O-PSM algorithm. At the same time, all the algorithms have not exhausted the budget because deciding whether to add users to the set depends on two factors, the budget and the marginal value of the remaining users. This can be seen from formula $b_i \leq \frac{B}{2}\frac{V_{i|A}}{V(A||i)}$ in line 2 of Algorithm 1. Similar phenomena also appeared in the following experiments. Notably, all algorithms have payments that decrease with increasing number of users, because when the user scale is small, the service provider has little choice of users. There are few users with a high marginal value density, which leads to the need for substantial fees when calculating the payment. As the number of users increases, the service providers have more cost-effective users to choose from, which reduces the cost of payment. Therefore, the greater the number of participating users is, the better the situation for the service provider.

Fig. 4c shows the number of winning users for the three algorithms. It can be seen that the number of winners of the

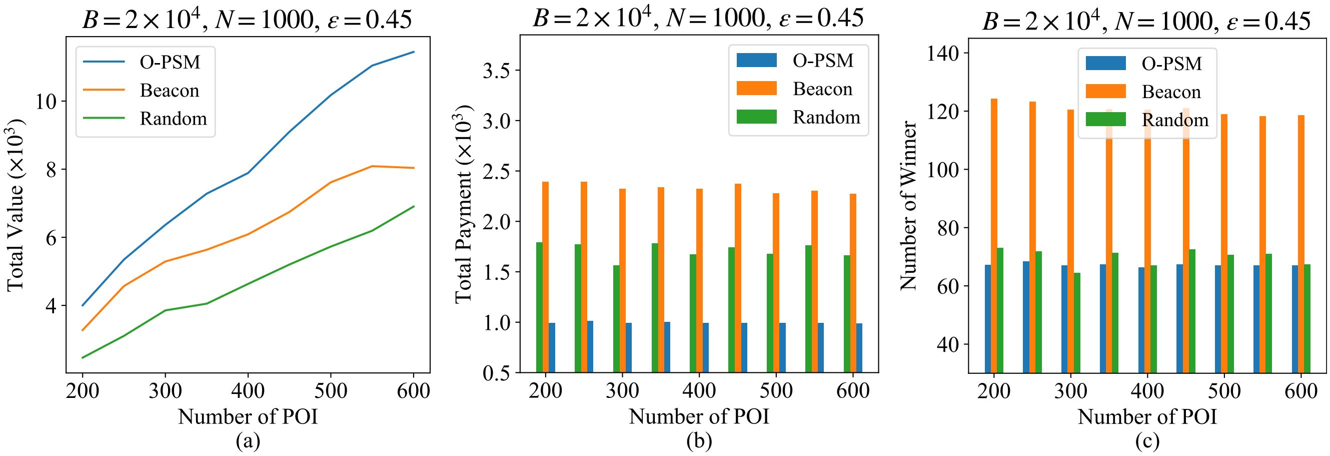


Fig. 6. Impact of the POI scale.

BEACON algorithm is large. This is because in the BEACON algorithm, the number of POIs covered by users is small, so more users need to be selected to obtain higher value, which is another reason for the high payment paid by the BEACON algorithm. Another point is the number of winners first increases and then stabilizes because the number of winners is limited by two main important factors, the budget of the service provider, that is, the total payment cannot exceed B, and the maximum coverage times of the POIs, that is, $\max\{r_m\}, m \in \mathcal{M}$. According to the algorithm, the winning user sequence must satisfy $|A| \le \max\{r_m\}$ because if $|A| = \max\{r_m\}$, then continuing to add new users to the user sequence does not increase the total value. According to the O-PSM algorithm and Fig. 4b, the total payment of the three algorithms is less than $B/2$, so the number of winners is affected more by $\max\{r_m\}, m \in \mathcal{M}$. Therefore, when the number of users is small, most users will be selected, but when the number of winning users reaches a certain scale, the diminishing marginal value begins to play a role, limiting the continued increase in winning users.

5.2.2 Impact of the Budget

In this experiment, the service provider's budget starts at 5,000 and gradually increases to 20,000 in steps of 1,000. The number of users is 1,000, and the number of POIs is 641. The experimental results are shown in Fig. 5.

Fig. 5a shows that as the budget gradually increases, the value curves of the three algorithms all show an increasing trend because when the budget is higher, there are more winning users, resulting in a larger total value. However, due to the characteristics of the diminishing marginal value, the marginal value of the last user who wins is lower, so the value curve does not substantially increase. As shown in Fig. 5b, as the budget gradually increases, the number of winning users increases; finally, the payment of the three algorithms gradually increases, but the payment of the O-PSM algorithm is still the smallest. The reasons O-PSM obtains a higher total value and a low payment are the same as those explained in 5.2.1, that is, the ordered-submodularity characteristics and long-tail effects. Fig. 5c shows the linear growth in the number of winning users. According to the algorithm allocation stage, as the budget gradually increases, the threshold $b_i \le (B/2)/(V_{i|A}/V(A\|i))$ also shows linear growth, resulting in linear growth of the number of winning users, and the reason why the BEACON algorithm has the most winners is the same as in 5.2.1. Notably, as the number of users linearly grows, the total value function continues to conform to the characteristics of diminishing marginal value.

5.2.3 Impact of the POI Scale

In this experiment, the number of POIs starts at 200 and gradually increases to 600 in steps of 50. The POI selection method is random selection. The number of users is 1,000, and the budget of the service provider is 20,000. The experimental results are shown in Fig. 6.

As shown in Fig. 6a, the total value rapidly increases with the linear increase in the number of POIs. This is because when the number of POIs increases, the same user can cover more POIs, increasing the value generated by each, so the total value rapidly grows. However, the growth rate of O-PSM is larger than that of the other algorithms because O-PSM calculates the total value based on the coverage probability, so a single user can generate more value at the POIs, even those POIs covered by a small probability. BEACON uses 0 and 1 to represent the coverage of POIs, so each user can cover a limited number of POIs (because for a single user, her/his active area is not large, which causes the BEACON algorithm to have many 0 values and few 1 values in the heatmap). Therefore, even as the number of POIs increases, the value users generate at new POIs is still limited.

Fig. 6b shows the payment of the three algorithms. The change in the number of POIs does not have a significant impact on the payment of the three algorithms because in the payment stage, the payment of the winning users is independent of the number of POIs. In the allocation stage of the O-PSM algorithm and the BEACON algorithm, the value of a user is determined by the marginal value density. Therefore, when the POI changes, the winning user sequence and total payment do not change significantly. Under the coverage probability model, regardless of the number of POIs, the judgment of cost-effective users is not greatly affected. In the allocation stage of the random algorithm, users are randomly selected regardless of the number of POIs. Fig. 6c shows the

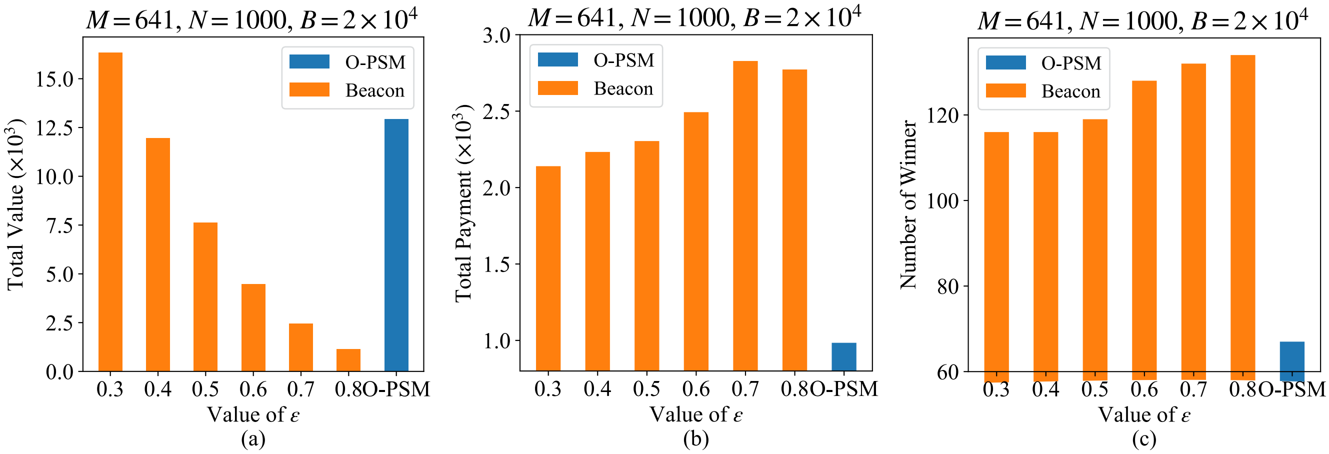


Fig. 7. Impact of ε on BEACON.

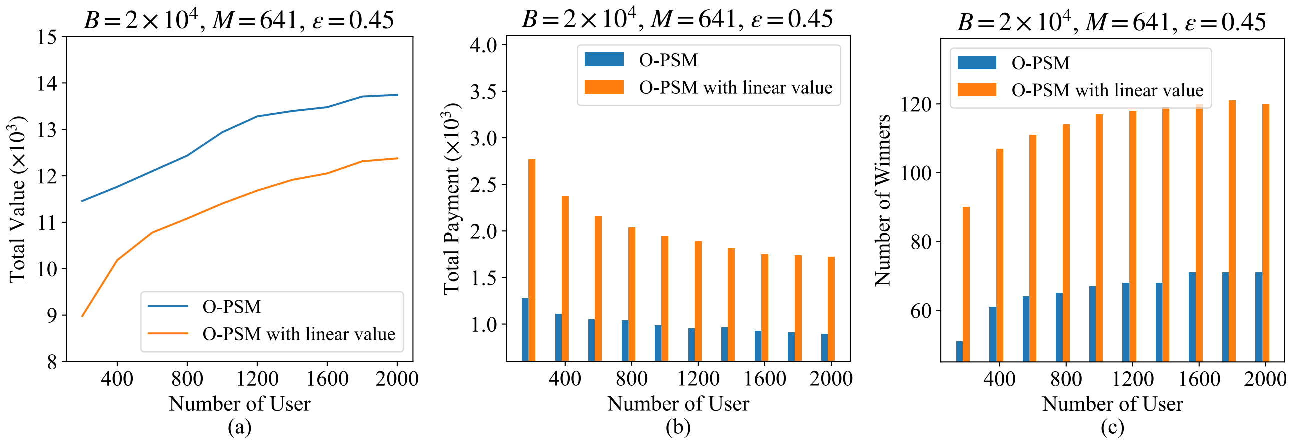


Fig. 8. Impact of Linear Value Function on O-PSM.

number of winning users; the number of winners for the three algorithms is relatively stable. The reason is the same as that in the analysis of Fig. 6b. Furthermore, O-PSM obtains greater value with lower payment due to the ordered submodularity characteristics and long-tail effect.

5.2.4 Impact of ε on BEACON

To refute the possibility that the performance of the BEACON algorithm is weakened, we conduct experiments on the impact of the value of ε on the performance of BEACON. In this experiment, the value of ε is increased from 0.3 to 0.8 in steps of 0.1. The number of users in the experiment is 1000, the number of POIs is 641, and the budget of the service provider is 20,000. The experimental results are shown in Fig. 7.

As shown in Fig. 7a, as ε increases, the total value obtained by the BEACON algorithm shows a decreasing trend because when ε increases, the number of POIs that a single user can cover rapidly decreases. Thus, the value that a single user can generate also decreases. Figs. 7b and 7c show that the BEACON algorithm payment and the number of winners increases as ε increases. This is because when the number of POIs covered by a single user decreases, affected by $\max\{r_m\}$, more users must be selected to achieve a better coverage rate for the POIs. Notably, when ε is less than or equal to 0.4, the total value of BEACON is greater than that of O-PSM, but in actual situations, for POIs with a user coverage probability less than 0.4, judging whether the user can cover them is difficult, and this article chooses 0.45 as a threshold, which is acceptable for the BEACON algorithm.

5.2.5 Impact of the Linear Value Function on O-PSM

In this experiment, we compare the O-PSM algorithm with the algorithm *O-PSM with linear value* using

$$V^{(m)}(\mathcal{U}) = v_m \frac{\sum_{i=1}^{\min\{r_m, |\mathcal{U}|\}} p_{im}}{r_m}, \tag{16}$$

as the value function. Formula (16) is a linear value function that accumulates the coverage probabilities of users for POI m, which is similar to the value function used in literature [5], [6]. It can be seen from Fig. 8a that O-PSM has a higher value thanks to the characteristic of diminishing marginal utility; meanwhile, Figs. 8b and 8c show that if a linear value function is adopted, the service provider will choose more users and pay more at the same time. Therefore, the crowdsensing service providers must consider that the information collected first has greater value when designing the value function.

5.2.6 Truthfulness Verification of O-PSM

This experiment verifies the truthfulness of O-PSM from two perspectives: 1) the bid of a winning user is changed to

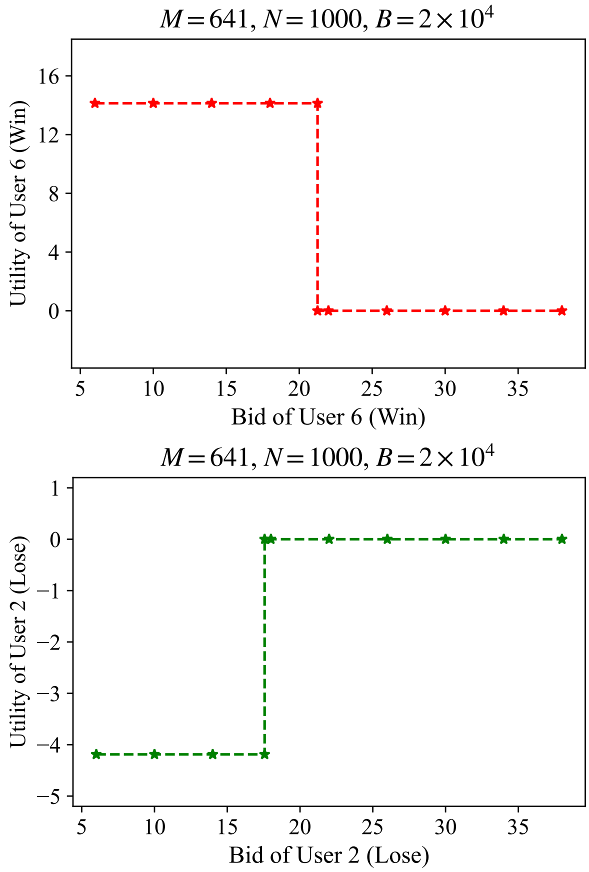


Fig. 9. Truthfulness verification.

observe the utility changes and 2) the bid of a losing user is changed to observe the utility changes. In this experiment, the number of users is 1,000, the budget of the service provider is 20,000, and the number of POIs is 641. The experimental results are shown in Fig. 9.

Fig. 9a shows the situation for winning user 6. Her/his truthful bid is 7.12, and when she/he wins, the service provider pays him/her 21.26, and her/his utility is 14.14. By constantly changing her/his bid, as long as the bid is lower than 21.26, the user can still win, but her/his utility remains unchanged at 14.14; this is because if the user can win, changing her/his bid will not affect the critical value; thus, the payment never changes, and the utility remains the same. When the bid exceeds 21.26, allocation fails, so the utility is 0.

Fig. 9b shows the situation for losing user 2. Her/his truthful bid is 21.76, the final payment is 0, and the utility is 0. When the bid is lower than 17.56, the user wins in the allocation stage, but the utility is 17.56-21.76= -4.2. When the bid exceeds 17.56, the user loses the allocation, and the payment and utility are still 0. These two examples illustrate that users cannot obtain greater utility by changing their bids, thus verifying the truthfulness of O-PSM.

6 Conclusion

The reverse auction mechanism is a good method to solve the mobile crowdsensing problem, but the main issue is that the service provider assumes that the user will be successful in collecting information in a specific area, which is not true in reality and violates the truthfulness of the mechanism. This article converts the coverage requirements submitted by users into a user coverage probability model, which effectively avoids this problem. By introducing ordered submodularity theory, a corresponding mechanism is designed to successfully solve this problem. Ordered submodularity theory is a new branch of submodular theory that can be used to construct value models described by probabilities to solve practical problems when combined with a mechanism design. We hope that this article provides a new development direction for research and application. However, many challenges remain, such as the integration of online environments and how service providers price POI. These problems are closer to the actual situation and are the goals of our future research.

References

[1] X. Kong, X. Liu, B. Jedari, M. Li, L. Wan, and F. Xia, "Mobile crowdsourcing in smart cities: Technologies, applications, and future challenges," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8095–8113, Oct. 2019.

[2] Y. Cai, G. Fragkos, E. E. Tsiropoulou, and A. Veneris, "A truth-inducing sybil resistant decentralized blockchain oracle," in *Proc. 2nd Conf. Blockchain Res. Appl. Innov. Netw. Serv.*, 2020, pp. 128–135.

[3] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2419–2465, Third Quarter 2019.

[4] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011.

[5] D. Zhao, X.-Y. Li, and H. Ma, "Budget-feasible online incentive mechanisms for crowdsourcing tasks truthfully," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 647–661, Apr. 2016.

[6] Z. Zheng, F. Wu, X. Gao, H. Zhu, S. Tang, and G. Chen, "A budget feasible incentive mechanism for weighted coverage maximization in mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 16, no. 9, pp. 2392–2407, Sep. 2017.

[7] Y. Zhan, Y. Xia, J. Zhang, and Y. Wang, "Incentive mechanism design in mobile opportunistic data collection with time sensitivity," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 246–256, Feb. 2018.

[8] F. Yucel and E. Bulut, "Online stable task assignment in opportunistic mobile crowdsensing with uncertain trajectories," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 9086–9101, Jun. 2022.

[9] J. Kleinberg, E. Ryu, and É. Tardos, "Ordered submodularity and its applications to diversifying recommendations," 2022, *arXiv: 2203.00233*.

[10] Y. Singer, "Budget feasible mechanisms," in *Proc. IEEE 51st Annu. Symp. Found. Comput. Sci.*, 2010, pp. 765–774.

[11] C. Yi and J. Cai, "Ascending-price progressive spectrum auction for cognitive radio networks with power-constrained multiradio secondary users," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 781–794, Jan. 2018.

[12] A. Chouayakh, A. Bechler, I. Amigo, L. Nuaymi, and P. Maillé, "Multi-block ascending auctions for effective 5G licensed shared access," *IEEE Trans. Mobile Comput.*, vol. 21, no. 11, pp. 4051–4063, Nov. 2022.

[13] J. Zhang, X. Yang, N. Xie, X. Zhang, A. V. Vasilakos, and W. Li, "An online auction mechanism for time-varying multidimensional resource allocation in clouds," *Future Gener. Comput. Syst.*, vol. 111, pp. 27–38, 2020.

[14] X. Liu, W. Li, and X. Zhang, "Strategy-proof mechanism for provisioning and allocation virtual machines in heterogeneous clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 7, pp. 1650–1663, Jul. 2018.

[15] J. Zhang, N. Xie, X. Zhang, and W. Li, "An online auction mechanism for cloud computing resource allocation and pricing based on user evaluation and cost," *Future Gener. Comput. Syst.*, vol. 89, pp. 286–299, 2018.

[16] J. Zhang, L. Chi, N. Xie, X. Yang, X. Zhang, and W. Li, "Strategy-proof mechanism for online resource allocation in cloud and edge collaboration," *Computing*, vol. 104, no. 2, pp. 383–412, 2022.

[17] D. T. Nguyen, L. B. Le, and V. Bhargava, "Price-based resource allocation for edge computing: A market equilibrium approach," *IEEE Trans. Cloud Comput.*, vol. 9, no. 1, pp. 302–317, First Quarter 2021.

[18] J. Zhang, W. Lou, H. Sun, Q. Su, and W. Li, "Truthful auction mechanisms for resource allocation in the Internet of Vehicles with public blockchain networks," *Future Gener. Comput. Syst.*, vol. 132, pp. 11–24, 2022.

[19] Y. Jiao, P. Wang, D. Niyato, and K. Suankaewmanee, "Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 9, pp. 1975–1989, Sep. 2019.

[20] N. Anari, G. Goel, and A. Nikzad, "Mechanism design for crowdsourcing: An optimal 1–1/e competitive budget-feasible mechanism for large markets," in *Proc. IEEE 55th Annu. Symp. Found. Comput. Sci.*, 2014, pp. 266–275.

[21] Z. Zheng, S. Yang, J. Xie, F. Wu, X. Gao, and G. Chen, "On designing strategy-proof budget feasible online mechanisms for mobile crowdsensing with time-discounting values," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 2088–2102, Jun. 2022.

[22] J. Hu, K. Yang, K. Wang, and K. Zhang, "A blockchain-based reward mechanism for mobile crowdsensing," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 1, pp. 178–191, Feb. 2020.

[23] K. Han, H. Liu, S. Tang, M. Xiao, and J. Luo, "Differentially private mechanisms for budget limited mobile crowdsourcing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 934–946, Apr. 2019.

[24] Y. Wang, Z. Cai, Z.-H. Zhan, Y.-J. Gong, and X. Tong, "An optimization and auction-based incentive mechanism to maximize social welfare for mobile crowdsourcing," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 3, pp. 414–429, Jun. 2019.

[25] H. Ma, D. Zhao, and P. Yuan, "Opportunities in mobile crowd sensing," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 29–35, Aug. 2014.

[26] Y. Zhan, Y. Xia, Y. Liu, F. Li, and Y. Wang, "Incentive-aware time-sensitive data collection in mobile opportunistic crowdsensing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 7849–7861, Sep. 2017.

[27] F. Yucel, M. Yuksel, and E. Bulut, "Coverage-aware stable task assignment in opportunistic mobile crowdsensing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3831–3845, Apr. 2021.

[28] S. Dobzinski, N. Nisan, and M. Schapira, "Approximation algorithms for combinatorial auctions with complement-free bidders," *Math. Operations Res.*, vol. 35, no. 1, pp. 1–13, 2010.

[29] S. Dobzinski, "Two randomized mechanisms for combinatorial auctions," in *Proc. Int. Workshop Approximation Randomization Combinatorial Optim.. Algorithms Techn.*, 2007, pp. 89–103.

[30] S. Assadi and S. Singla, "Improved truthful mechanisms for combinatorial auctions with submodular bidders," in *Proc. IEEE 60th Annu. Symp. Found. Comput. Sci.*, 2019, pp. 233–248.

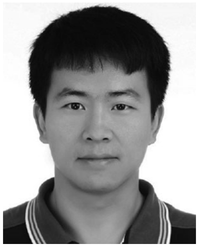
[31] Y. Jiao, P. Wang, D. Niyato, B. Lin, and D. I. Kim, "Toward an automated auction framework for wireless federated learning services market," *IEEE Trans. Mobile Comput.*, vol. 20, no. 10, pp. 3034–3048, Oct. 2021.

[32] L. Mashayekhy, M. Nejad, D. Grosu, and A. Vasilakos, "An online mechanism for resource allocation and pricing in clouds," *IEEE Trans. Comput.*, vol. 65, no. 4, pp. 1172–1184, Apr. 2016.

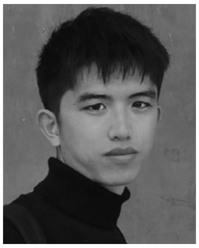
[33] R. B. Myerson, "Optimal auction design," *Math. Operations Res.*, vol. 6, no. 1, pp. 58–73, 1981.

[34] A. Mu'Alem and N. Nisan, "Truthful approximation mechanisms for restricted combinatorial auctions," *Games Econ. Behav.*, vol. 64, pp. 612–631, 2008.

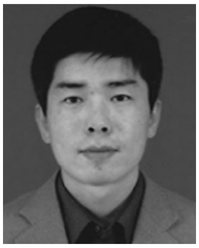
[35] T. Nisan, E. Roughgarden, E. Tardos, and V. Vazirani, *Algorithmic Game Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2007.



Jixian Zhang received the MS degree in computer science from the University of Electronic Science and Technology of China, in 2006, and the PhD degree in computer science from the University of Electronic Science and Technology of China, in 2010. He is an associate professor with the School of Computer Science and Engineering, Yunnan University. He has published more than 30 articles in peer-reviewed journals and conferences, e.g., *Future Generation Computer Systems*, *Journal of Obstetrics and Gynaecology Canada*, *Cluster Computing*, *Computing* and CMC. His research interests include cloud computing, edge computing and mechanism design.



Yi Zhang is currently working toward the master's degree from Yunnan University and is expected to received the master's degree, in 2023. His main study and research directions include resource allocation, deep learning, vehicular communication, etc.



Hao Wu received the PhD degree in computer science from the Huazhong University of Science and Technology, in 2007. Currently, he is an associate professor with the School of Information Science and Engineering, Yunnan University, China. He has published more than 60 article in peer-reviewed journals and conferences, e.g., *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Services Computing*, NeuNet, WWWJ, *Future Generation Computer Systems*, *Knowledge-Based Systems*, *Personal and Ubiquitous Computing*, *Automated Software Engineering*, ICWS, and ICSOC. He has coauthored two monographs published in World Scientific. His research interests include service computing, information filtering, and recommender systems.

Weidong Li received the PhD degree from the Department of Mathematics, Yunnan University, in 2010. He is currently a professor with the School of Mathematics and Statistics, Yunnan University. He has published more than 90 articles in peer-reviewed journals and conferences, e.g., *IEEE Transactions on Parallel and Distributed Systems*, ALGO, JOA, *Theoretical Computer Science*, *Journal of Combinatorial Optimization*, *Future Generation Computer Systems*, and *Journal of Obstetrics and Gynaecology Canada*. His main research interests include combinatorial optimization, approximation algorithms, randomized algorithms and cloud computing.