Towards Demand-Driven Dynamic Incentive for Mobile Crowdsensing Systems

Jiahui Hu, *Member, IEEE*, Zhibo Wang, *Senior Member, IEEE*, Jian Wei, Ruizhao Lv, Jing Zhao, Qian Wang, *Senior Member, IEEE*, Honglong Chen, *Member, IEEE*, and Dejun Yang, *Senior Member, IEEE*

***Abstract*— Incentive mechanisms have been commonly proposed to encourage people to participate in mobile crowdsensing (MCS). However, most of them set unchangeable rewards for sensing tasks, while the inherent inequality and on-demand feature of sensing tasks have been long ignored, especially for location-dependent sensing tasks (LDSTs). In this paper, we focus on location-dependent MCS systems and propose a demand-driven dynamic incentive mechanism that dynamically changes the rewards of sensing tasks at each sensing round in an on-demand way to balance their popularity. A demand indicator is introduced to characterize the demand of each sensing task by considering its deadline, completing progress, and number of potential participants. At each sensing round, we use the Analytic Hierarchy Process (AHP) to calculate the relative demands of all sensing tasks and then determine their rewards accordingly. Moreover, we consider two task selection problem with participatory users and opportunistic users, respectively, and prove that both of them are NP-hard. We propose an optimal dynamic programming based solution for participatory scenario and an optimal backtracking based solution for opportunistic scenario to help each user select tasks while maximizing its profit. Extensive experiments show that the demand-driven dynamic incentive mechanism outperforms existing incentive mechanisms.**

***Index Terms*— Mobile crowdsensing, dynamic incentive, task allocation, pay on-demand.**

I. Introduction

WITH the rapid development of technology, mobile devices (e.g., smartphones) become more and more powerful in sensing as they are equipped with a rich set of embedded sensors (e.g., camera, microphone, and GPS). Nowadays a mobile user carrying a mobile device is not only a human but has become a powerful mobile sensing platform that can sense environments as well as people's behaviors. This fact has benefited the emergence of MCS systems, such as Waze, which leverage the power of large number of mobile users to collect data for sensing applications instead of using traditional sensors. A typical crowdsensing system [1] consists of a cloud server and a large number of mobile users where the cloud server publishes sensing tasks and mobile users use their mobile devices to collect sensing data to complete the published tasks. Thanks to the mobility of mobile users and the popularity of mobile devices, crowdsensing has become an effective technique to collect massive data for lots of sensing applications, and it is especially suitable for user-centric and location-dependent sensing applications.

Nowadays, the crowdsensing technology has been adopted by several sensing applications to recruit mobile users for massive location-dependent data collection, such as traffic condition monitoring [2], air quality monitoring [3], and noise pollution assessment [4]. To perform the LDSTs, mobile users should arrive at a specific location and contribute the location related sensing data to the cloud server. When mobile users contribute sensing data in crowdsensing, both time and physical resources are spent to complete sensing tasks. Thus, mobile users have no motivation to participate in crowdsensing without an appropriate incentive. Moreover, privacy-sensitive mobile users may be further prevented from contributing sensing data due to the privacy leakage concerns. Recently many incentive mechanisms have been proposed to improve users' participation. Some of them are based on game-theoretic technologies that allocate tasks to mobile users with the objective of maximizing the social surplus [5]–[9]. Some of them designed quality-orientated incentive mechanisms to improve the quality of sensing data [10]–[13]. Moreover, with the increase of location-dependent applications, incentive mechanisms for location-dependent crowdsensing systems are proposed in [11], [13]–[17].

It is worth noting that most of existing incentive mechanisms set unchangeable/fixed rewards for sensing tasks, although different rewards may be given to different tasks, where the reward of a sensing task does not change once it is initially determined. This however is not suitable for location-dependent crowdsensing systems since the location becomes another important factor besides the reward influencing the decision of users to perform a sensing task or not. In this

Manuscript received September 3, 2019; revised January 19, 2020; accepted April 12, 2020. Date of publication April 23, 2020; date of current version July 10, 2020. This work was supported in part by the National Natural Science of China under Grant 61872274, Grant 61822207, Grant U1636219, and Grant 61772551, in part by the Equipment Pre-Research Joint Fund of Ministry of Education of China (Youth Talent) under Grant 6141A02033327, in part by the National Key Research and Development Program of China under Grant 2019YFA0706403, in part by the Natural Science Foundation of Hubei Province under Grant 2017CFB503 and Grant NSF 1717315, and in part by the Fundamental Research Funds for the Central Universities under Grant 2042018gf0043, Grant 2042019gf0098, and Grant 18CX07003A. The associate editor coordinating the review of this article and approving it for publication was M. Li. *(Corresponding author: Zhibo Wang.)*

Jiahui Hu, Zhibo Wang, Jian Wei, Ruizhao Lv, Jing Zhao, and Qian Wang are with the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China (e-mail: jiahuihu@whu.edu.cn; zbwang@whu.edu.cn; wj9528@whu.edu.cn; ruizhaolv@whu.edu.cn; zhaojing14@whu.edu.cn; qianwang@whu.edu.cn).

Honglong Chen is with the College of Control Science and Engineering, China University of Petroleum, Qingdao 266580, China (e-mail: chenhl@upc.edu.cn).

Dejun Yang is with the Department of Computer Science, Colorado School of Mines, Golden, CO 80401 USA (e-mail: djyang@mines.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2020.2988271

paper, we argue that inherent inequality exists among LDSTs and the demands of tasks for participants change dynamically over time. Generally speaking, compared to the remote tasks with low rewards, mobile users prefer to perform close tasks with high rewards, leading to the unbalanced completion of tasks. That is, the popularity of tasks will differ from each other because of the location difference. Hence in fixed incentive mechanisms, the location of each task and its initial reward inherently determine its popularity from the beginning, leading to a low coverage issue that only popular tasks can be completed while unpopular LDSTs cannot be completed on time. This problem motivates us to design a dynamic incentive mechanism that dynamically changes the reward of each sensing task based on the real-time demands of tasks to balance the popularity of sensing tasks so that even far away tasks can also be completed before their deadlines.

Note that there also exist some dynamic incentive mechanisms [9], [11], [13]. In [9], to maintain adequate participation level, the authors introduced a reverse auction-based dynamic pricing incentive mechanism for participatory sensing. However, they did not consider the location difference and demand difference. Guo et al. focused on data quality and proposed a dynamic incentive mechanism, where the authors set different/dynamic budget value for each sensing task based on the spatio-temporal popularity level [13]. However, the proposed mechanism only considered one-shot sensing tasks that can be completed by one-time measurement and tasks needing multiple measurements have been ignored. Although there are different reward budgets between different tasks, it still would not be changed once determined initially. Therefore, it can be considered as a fixed incentive mechanism with different budgets for each task. In [11], the authors proposed a steered incentive mechanism where the points can be changed in every session so that the quality of service rather than data size can be improved. However, the points decrease over time, which will discourage users and results in less engagement of participants. Moreover, they did not consider the difference of deadlines between different LDSTs.

In this paper, we focus on location-dependent MCS systems with the Worker Selected Tasks (WST) mode. In contrast to the Server Assigned Tasks (SAT) mode, the WST mode are commonly used by many popular crowdsensing applications, such as Gigwalk and FieldAgent. Instead of allocating tasks to mobile users by the server in a centralized manner, it is more practical that mobile users select tasks in a distributed way. In our system, the server only needs to publish tasks with rewards at each sensing round, and then mobile users select a set of tasks to be performed according to their cost and time budget. Note that the complicated negotiation process can be avoided between the server and mobile users.

We propose a demand-driven dynamic incentive mechanism to encourage mobile users to participate in crowdsensing*. Instead of using a fixed reward for a task all the time, we argue that the reward should be paid on-demand and changes dynamically at each sensing round. Intuitively, the closer to the deadline or the smaller completing progress or the less mobile users around a task, the larger reward is expected to improve the task's popularity and attraction. Thus, we introduce the demand indicator to characterize the demand of each sensing task which takes several factors into consideration, such as the deadline, the completing progress and the number of potential participants of a sensing task. At each sensing round, AHP is adopted to model and calculate the relative demands of all sensing tasks and then their rewards can be determined accordingly.

We summarize our contributions as follows.

- We propose a demand-driven dynamic incentive mechanism for location-dependent MCS systems, which provides a concrete guideline on how to dynamically change the reward of each sensing task according to its real-time demand.
- We propose a demand indicator to characterize the demand of each sensing task by taking important factors into consideration, and adopt the Analytic Hierarchy Process to model and calculate the relative demands of all sensing tasks.
- We prove that the task selection problem under two different movement patterns of users (e.g., participatory and opportunistic) are NP-hard. An optimal dynamic programming solution and an optimal backtracking based solution are further proposed to help mobile users select optimal set of tasks while maximizing their profits at each sensing round.
- We conduct extensive experiments to compare the proposed demand-driven dynamic incentive mechanism with existing incentive mechanisms by simultaneously considering participatory and opportunistic users. The experimental results show that the proposed mechanism achieves better participation balance among tasks.

The remainder of this paper is organized as follows. The existing incentive mechanisms are briefly discussed for crowdsensing systems in Section II. We introduce the system overview and describe the task selection and incentive design problems in Section III. We present the demand-driven dynamic incentive mechanism in Section IV, and two distributed task selection algorithms in Section V. We evaluate the performance of the proposed algorithms in Section VI and finally conclude the paper in Section VII.

II. Related Work

In recent years, benefit from a rich set of embedded sensors, location-dependent incentive mechanisms have drawn great attention. In [19], the location-dependent crowdsensing problems are classified into two modes: WST and SAT.

In the SAT mode, the server has the global information of the tasks as well as mobile users, and usually assign tasks to mobile users using the auction-based mechanisms. In [20], the authors applied reverse auctions in the economic field to the crowdsensing incentive mechanisms. This application not only minimized the payment cost but also ensured the high participation of users relatively. In [21], a double auction mechanism is presented for motivating participants

*A preliminary version of this work was published in IEEE International Conference on Distributed Computing Systems (ICDCS'18) [18].

to join the K anonymity of location-sensitive. Based on the reverse combinatorial auction model, Feng et al. [14] proposed a truthful mechanism to motivate the participants. In [22], the authors applied the multi-attribute auction mechanism to reverse auction, which took both the participation rate of users and the quality of sensing data into consideration. [23] designed a full-pay auction method to motivate participants to participate in, of which only the bidder who contributes mostly can get the payoff. In [15], the authors proposed the VCG auction mechanism and designed an updating rule for online crowdsensing incentive mechanism in order to achieve the social welfare benefits maximization.

For the WST mode, mobile users can select any tasks autonomously without contacting with the server. Although it can hardly achieve the maximization objective as the WST mode, it is the typical mode used in many popular crowdsensing systems such as Gigwalk, Amazon Mechanical Turk, and Field Agent, actually. Besides, in [11], the authors proposed steered crowdsensing. It used the game elements on location-based services to control the incentives of participants. In [24], the authors designed an asynchronous and distributed task selection algorithm. With the help of this algorithm, mobile users can find the best schedule. Compared to the SAT mode, workers in the WST mode can submit less personal information and it can improve the participation of workers. Moreover, the WST mode's procedure is relatively concise. However, some sensing tasks may not be completed in this mode, while others are completed redundantly since the server does not have any control over the allocation of sensing tasks.

Recently some works considered the user privacy leakage and data security in data collection and analysis [25]–[29]. Considering that the leakage of true locations to the server may be harmful to users, some work begin to address the problem of task allocation in mobile crowdsensing with location privacy protection [30].

In this paper, we address the WST mode in location-dependent crowdsensing systems and design a demand-driven dynamic incentive mechanism that can change the rewards of sensing tasks in an on-demand way dynamically.

III. System Overview and Problem Statement

In this section, we first present the high-level overview of location-dependent crowdsensing systems with the dynamic incentive mechanism, and then describe the location-dependent dynamic incentive design problem and the distributed task selection problem.

A. System Overview

We consider the location-dependent crowdsensing applications which leverage the power of the crowd to collect massive sensing data. In particular, we take the noise pollution assessment as an example for crowdsensing applications, which aims to provide the accurate noise pollution levels of different regions in a city to the public. It is expensive and time-consuming to deploy specific equipments to measure noise pollution levels considering the large-scale of a city. Even the equipments are deployed, they can only provide a coarse-grained noise measurement of the city. In contrast, we can use

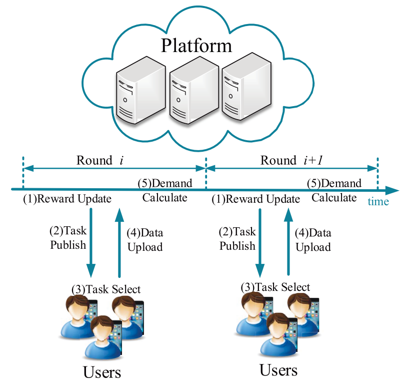


Fig. 1. The architecture of crowdsensing systems with the dynamic incentive mechanism.

the idea of crowdsensing that leverages the power of the crowd to realize cheap and fine-grained noise measurements. Each participant can use its mobile device to measure the noise, so there is no need to deploy expensive and specific equipments. The participants can move to the specified places to make quick and convenient measurements, which can realize fine-grained noise measurements.

Figure 1 shows the architecture of crowdsensing systems with the proposed dynamic incentive mechanism. The platform publishes a set of sensing tasks to mobile users and provides rewards for tasks to incentivize mobile users to accomplish tasks. Different from crowdsensing systems with the SAT mode, each mobile user in our crowdsensing systems with the WST mode does not need to send its bid to the platform to compete tasks. Instead, a mobile user can select a set of tasks to perform in a distributed way according to its time budget and cost consumption. We assume all mobile users are rational so they would not perform a task if the cost spent is larger than the gained reward or the time budget is not satisfied.

In this paper, we propose a novel demand-driven dynamic incentive mechanism for location-dependent MCS systems. As shown in Figure 1, the data collection process is divided into multiple sensing rounds. At each sensing round, mobile users select tasks, perform the selected tasks and upload the sensing data to the platform. The platform collects the sensing data and calculates the demands of all sensing tasks. In the next sensing round, the platform updates the reward for each sensing task and publishes the tasks with updated rewards to the mobile users. The task selection process for each mobile user and the rewards update process on the platform continues repeatedly until all the tasks are completed. After receiving the sensing data of a task from mobile users, the platform aggregates the sensing data to make an estimate. If all the sensing data are from the same mobile user, the estimate may be biased or cannot be trusted. In order to guarantee the sensing quality of each task, we assume that each task requires independent sensing measurements from multiple mobile users.

Besides, we consider the task selection problem under two different user movement scenarios: the participatory pattern and the opportunistic pattern. The participatory movement pattern is widely used in MCS systems that users do not care the destination and just want to earn rewards by performing tasks. The opportunistic movement pattern is also popular for users with daily route (e.g., leave from the home and arrive at the office) so they can perform tasks on the way to the destination as the time allows.

B. Location-Dependent Dynamic Incentive Problem

The platform expects each sensing task to be completed before its deadline, and provides rewards to encourage mobile users to participate in MCS. We assume the platform has a total budget $\mathcal{B}$ for all the sensing tasks, and the total rewards paid to mobile users cannot exceed $\mathcal{B}$. However, existing incentive mechanisms mainly apply unchangeable rewards for sensing tasks, which have several drawbacks. First, it is difficult or impossible to decide the optimal reward for each sensing task. If the rewards are set too high, the platform is harmed as its welfare is small or be negative, while if the rewards are set too small, there may not be enough participants to complete sensing tasks. Second, it may lead to the problem that some sensing tasks cannot be completed before their deadlines. It is possible that some sensing tasks are not popular to mobile users because they are in remote places or their rewards are small. The popularity cannot be changed if the rewards are fixed, and therefore these sensing tasks cannot be completed on time.

To solve these issues, we propose to dynamically change the reward of each sensing task to balance the popularity of sensing tasks in an on-demand way. The dynamic incentive mechanism needs to satisfy two objectives. First, each location-dependent sensing task should be completed before its deadline. Second, the welfare of the platform should be as large as possible. Therefore, the problem is how to characterize the demand of location-dependent sensing tasks and dynamically change the rewards of sensing tasks to realize these two objectives. We call the problem as the location-dependent dynamic incentive problem.

C. Location-Dependent Task Selection Problem

At each sensing round, the platform publishes a set of sensing tasks with rewards to mobile users, and each mobile user can choose to perform a set of tasks according to its time budget and cost consumption. Let $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ denote the set of sensing tasks where t_i denote the ith task. Let $\mathcal{U} = \{u_1, u_2 \ldots, u_n\}$ denote the set of mobile users where u_i is the ith mobile user. Each sensing task is location-dependent which means that each sensing task t_i is associated with a specific location L_{t_i}. We also assume that each sensing task t_i is associated with a deadline D_{t_i} that the task is expected to be completed before the deadline. Each task t_i requires φ_i mobile users to contribute sensing data and each mobile user contributes sensing data to each sensing task t_i at most once. The reward of a sensing task changes at each round. We use $r^k_{t_i}$ to denote the reward of task t_i at the kth round. Then we will discuss the task selection problem under the participatory and opportunistic movement patterns, respectively.

1) Participatory Task Selection Problem: In this scenario, each user does not have a destination to arrive at. At the beginning, a user needs to upload its location and time budget to the platform. Let $\mathcal{T}^k_{u_i}$ denote the set of tasks chosen by user u_i and $B^k_{u_i}$ denote the time budget of user u_i at the kth round.The time spent for completing multiple tasks is comprised of two parts: the time for traveling multiple locations associated with the selected tasks, and the time for data sensing at each location. Usually the latter is negligible compared to the former. Thus, we let the time spent for completing multiple tasks to be the time spent for traveling multiple locations associated with the selected tasks, denoted by $\Gamma_{\mathcal{T}^k_{u_i}}$. Since each mobile user has a time budget, $\Gamma_{\mathcal{T}^k_{u_i}}$ should be no larger than $B^k_{u_i}$.

At the kth sensing round, the participatory task selection problem for the mobile user u_i can be formulated as follows:

$$\begin{aligned} \max \quad & P(\mathcal{T}^k_{u_i}) = \sum_{t_j \in \mathcal{T}^k_{u_i}} r^k_{t_j} - C(\mathcal{T}^k_{u_i}) \\ \text{s.t.} \quad & \Gamma_{\mathcal{T}^k_{u_i}} \leq B^k_{u_i} \end{aligned} \tag{1}$$

where $r^k_{t_j}$ denotes the reward of task t_j at the kth round, and $C(\mathcal{T}^k_{u_i})$ denotes the minimum cost spent to perform the set of tasks $\mathcal{T}^k_{u_i}$, which is proportional to the minimum traveling distance from the original location of mobile user u_i to all the locations of tasks in $\mathcal{T}^k_{u_i}$. $P(\mathcal{T}^k_{u_i})$ denotes the total profit received by u_i for performing tasks in $\mathcal{T}^k_{u_i}$, which is the difference between the total rewards received by u_i ($\sum_{t_j \in \mathcal{T}^k_{u_i}} r^k_{t_j}$) and the minimum cost ($C(\mathcal{T}^k_{u_i})$).

As presented in Eq. 1, the objective of the participatory task selection problem for u_i at the kth round is to maximize its total profit, while the constraint indicates that the total traveling time should be no larger than user's time budget.

2) Opportunistic Task Selection Problem: In this scenario, users perform tasks on the way to their destinations (e.g., on the way to the office). Each user needs to upload its current location, the destination and the time budget to the platform. Let $\widetilde{\Gamma}_{\mathcal{T}^k_{u_i}}$ denote the detour time from the start point to the destination for performing tasks in the selected task set $\mathcal{T}^k_{u_i}$.

At the kth sensing round, the opportunistic task selection problem for the mobile user u_i can be formulated as follows:

$$\begin{aligned} \max \quad & P(\mathcal{T}^k_{u_i}) = \sum_{t_j \in \mathcal{T}^k_{u_i}} r^k_{t_j} - \Delta C(\mathcal{T}^k_{u_i}) \\ \text{s.t.} \quad & \widetilde{\Gamma}_{\mathcal{T}^k_{u_i}} \leq B^k_{u_i} \end{aligned} \tag{2}$$

where $\Delta C(\mathcal{T}^k_{u_i})$ denotes the additional cost of detour path, which is the difference between the detour cost and the original cost.

IV. Demand-Driven Dynamic Incentive

At each sensing round, each mobile user chooses a set of tasks and reports its sensing results to the platform. Therefore, the platform is aware of the completing progress of all tasks at the end of each sensing round. The basic idea of our algorithm is to dynamically change the reward of each task based on the demand of each task.

We introduce a demand indicator to characterize the demand of each location-dependent sensing task. Let $\mathcal{D}_k = (d^k_1, d^k_2, \ldots, d^k_n)$ denote the demands of all sensing tasks at the kth sensing round, where d^k_i denotes the demand of the ith task at the kth round. The demand of a task can be determined by many factors, such as the deadline, the

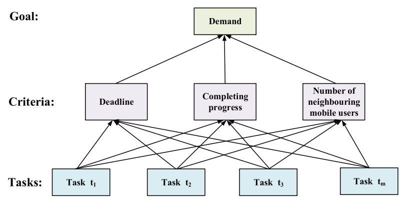


Fig. 2. The hierarchical structure of deciding the demands of tasks.

completing progress and the number of neighboring mobile users of a task. The user whose distances is less than R meters to a task is called a neighboring user of the task. Intuitively, the closer to the deadline, the larger the demand; the smaller the completing progress, the larger the demand; the less number of neighboring mobile users of a task, the larger the demand. Thus, we use the three factors to determine the demand of a task t_i.

$$d_i^k = w_1 X_{i_1}^k + w_2 X_{i_2}^k + w_3 X_{i_3}^k \tag{3}$$

where $X_{i_1}^k$, $X_{i_2}^k$ and $X_{i_3}^k$ represent the demands affected by the deadline, the completing progress, and the number of neighboring mobile users for task t_i, respectively. w_1, w_2 and w_3 are the weights to measure the relative importance of these three factors and we let $w_1 + w_2 + w_3 = 1$.

In our system, the rewards are given according to the demands. The higher the demand, the higher the reward. However, the absolute value of a demand actually does not have too much meaning, but instead the comparison of the demands of all sensing tasks are more important. This will help us use appropriate rewards to balance the popularity of sensing tasks. The Analytic Hierarchy Process (AHP) [31] is an effective model that combines qualitative and quantitative information to determine the relative ranking of alternatives (e.g., sensing tasks), and the ranking of criteria (e.g., three factors), which is a perfect model for our dynamic incentive problem.

Figure 2 shows the framework for our problem consisting of three levels, the alternative level, the criteria level and the goal level. The alternatives are the sensing tasks. The criteria are the demands of the deadline, the completing progress and the number of neighboring mobile users. The goal is to calculate the demands of all sensing tasks. In the following, we first quantify the demands of the three factors and use the AHP framework to calculate the demands of sensing tasks.

A. Demands of Three Factors

Demand affected by the deadline: Each sensing task is associated with a deadline and the required number of measurements are expected to be received before the deadline. The closer to the deadline, the higher demand will be required. Moreover, the closer to the deadline, the faster the growth rate of demand will be required. Therefore, the demand affected by the deadline is represented as follows:

$$X_{i_1}^k = \lambda_1 \ln(1 + \frac{1}{\tau_i - (k-1)}) \tag{4}$$

where τ_i is the deadline of task t_i, λ_1 is a coefficient that scales the value of the demand affected by the deadline. We can see that the demand $X_{i_1}^k$ increases as the round k approaches to the deadline of task t_i and is upper bounded by $\lambda_1 \ln 2$. Furthermore, the growth rate of demand $X_{i_1}^k$ increases as the round k approaches to the deadline.

Demand affected by the completing progress: The completing progress is another factor that can affect the demand of a task, which is defined as π_i/φ_i where π_i the number of received measurements and φ_i is the required number of measurements of task t_i. The larger the completing progress, the smaller demand will be required. Moreover, the larger the completing progress, the faster the reduction rate of demand will be required. Therefore, we have

$$X_{i_2}^k = \lambda_2 \ln(1 + (1 - \frac{\pi_i}{\varphi_i})) \tag{5}$$

where λ_2 is a coefficient that scales the value of the demand affected by the completing progress. We can see that the demand decreases as the completing progress increases and is lower bounded by 0. Furthermore, the reduction rate of demand $X_{i_2}^k$ increases as the completing progress approaches to 1.

Demand affected by the number of neighboring mobile users: Some tasks are surrounded by many mobile users, while some tasks are at far away locations with few neighboring mobile users. Mobile users would not select far away tasks only if high rewards are provided. Therefore, tasks with less neighboring mobile users should be given higher demands to increase their attractions to mobile users. Then we have

$$X_{i_3}^k = \lambda_3 \ln(1 + (1 - \frac{N_i}{N_{max}})) \tag{6}$$

where λ_3 is a coefficient that scales the value of the demand affected by the neighboring mobile users. N_i is the number of neighboring mobile users of task t_i, and $N_{max} = \max(N_i)$ is the maximum number of neighboring mobile users among all tasks. We can see that the less neighboring mobile users, the larger demand is required. The demand is lower bounded by 0 and upper bounded by $\lambda_3 \ln 2$.

B. Weights Calculation With AHP

Figure 2 shows the AHP framework for demand calculation. The demand of each sensing task can be calculated according to Eq. 3 where $X_{i_1}^k$, $X_{i_2}^k$ and $X_{i_3}^k$ are the three criteria C_1, C_2 and C_3 for tasks respectively, and $W = (w_1, w_2, w_3)^T$ is the vector of weights for criteria. In the following, we use the AHP to derive the appropriate values for the vector of weights.

TABLE I

AN EXAMPLE OF PAIRWISE COMPARISON MATRIX $A = (a_{ij})_{3\times 3}$

	C_1	C_2	C_3
C_1	1	3	5
C_2	1/3	1	2
C_3	1/5	1/2	1

Pairwise Comparison Matrix A: We use the pairwise comparison matrix $A = (a_{ij})_{3\times 3}$ to express the relative importance of one criteria over another. Generally, in practical the values in the matrix are always determined by experts and different for different application scenarios. For ease of understanding, we give an example like $A = (a_{ij})_{3\times 3}$. Each entry a_{ij} represents the relative importance of the criteria

TABLE II

NORMALIZED PAIRWISE COMPARISON MATRIX $\bar{A} = (\bar{a}_{ij})_{3\times 3}$ FOR THE EXAMPLE IN TABLE I

	C_1	C_2	C_3
C_1	0.652	0.667	0.625
C_2	0.217	0.222	0.250
C_3	0.131	0.111	0.125

C_i over the criteria C_j. If $a_{ij} > 1$, the criteria C_i is more important than the criteria C_j, while if $a_{ij} < 1$, the criteria C_i is less important than the criteria C_j. $a_{ij} = 1$ if the criteria C_i and C_j have the same importance. The entries a_{ij} and a_{ji} satisfy that $a_{ij} \times a_{ji} = 1$. In the AHP, the relative importance between two criteria is measured according to a numerical scale from 1 to 9 [31]. We can choose suitable values from 1 to 9 for a_{ij} according to the relative importance between two criteria in real scenarios.

Here we use an example in Table I to explain the pairwise comparison matrix. For example, $a_{12} = 3$ means the criteria C_1 (the deadline) is slightly more important than the criteria C_2 (the completing progress). $a_{13} = 5$ means the criteria C_1 (the deadline) is strongly more important than the criteria C_3 (the number of neighboring mobile users).

We then derive the normalized pairwise comparison matrix $\bar{A} = (\bar{a}_{ij})_{3\times 3}$ by normalizing A in each column. That is, each entry is calculated as $\bar{a}_{ij} = \frac{a_{ij}}{\sum_{k=1}^{3} a_{kj}}$. The normalized pairwise comparison matrix derived from Table I in shown in Table II.

Vector of weights: With the normalized pairwise comparison matrix, the vector of weights $W = (w_1, w_2, w_3)^T$ can be calculated by averaging the entries on each row of $\bar{A}$. That is,

$$w_i = \frac{1}{3}\sum_{j=1}^{3} \bar{a}_{ij} \tag{7}$$

Therefore, we can observe that the vector of weights $W = (0.648, 0.230, 0.122)^T$ for the example in Table II, which reflects the relative importance of the criteria on total demand. Since $0 \le X_{i_1}^k \le \lambda_1 \ln 2$, $0 \le X_{i_2}^k \le \lambda_2 \ln 2$ and $0 \le X_{i_3}^k \le \lambda_3 \ln 2$, and $w_1 + w_2 + w_3 = 1$, we can have $d_i^k = w_1 X_{i_1}^k + w_2 X_{i_2}^k + w_3 X_{i_3}^k \le \lambda_{max} \ln 2$ where $\lambda_{max} = \max(\lambda_1, \lambda_2, \lambda_3)$.

C. Demand Calculation and Reward Update

With the vector of weights and the demands affected by three factors, we can calculate the demands of all sensing tasks according to Eq. 3. That is, $d_i^k = w_1 X_{i_1}^k + w_2 X_{i_2}^k + w_3 X_{i_3}^k$. We then normalize the demand d_i^k to a scale $[0, 1]$. Since $0 \le X_{i_1}^k \le \lambda_1 \ln 2$, $0 \le X_{i_2}^k \le \lambda_2 \ln 2$ and $0 \le X_{i_3}^k \le \lambda_3 \ln 2$, and $w_1 + w_2 + w_3 = 1$, we can have $d_i^k \le \lambda_{max} \ln 2$ where $\lambda_{max} = \max(\lambda_1, \lambda_2, \lambda_3)$. Therefore, the normalized demand $\bar{d}_i^k$ can be calculated by $\bar{d}_i^k = \frac{d_i^k}{\lambda_{max} \ln 2}$.

We map the normalized demands into N levels and assign the reward to a sensing task according to its demand level. Table III shows an example of $N = 5$ demand levels. The demand level of a task is 2 if its normalized demand falls in $(0.2, 0.4]$.

TABLE III

AN EXAMPLE OF DEMAND LEVELS WHEN $N = 5$

Demand	[0, 0.2]	(0.2, 0.4]	(0.4, 0.6]	(0.6, 0.8]	(0.8, 1.0]
Level	1	2	3	4	5

We then determine the reward of the task according to its demand level by using the following rule.

$$r_{t_i}^k = r_0 + \lambda(DL_{t_i}^k - 1) \tag{8}$$

where $r_{t_i}^k$ is the updated reward for sensing task t_i at the kth sensing round, r_0 is the reward associated with the demand level 1 and $DL_{t_i}^k$ is the demand level of sensing task t_i at the kth sensing round. We can see that the reward increases linearly as the demand level increases and λ is the increasing scale. The maximum reward one can obtain for one measurement is $r_0 + \lambda(N-1)$. Considering that each task t_i requires φ_i measurements, the maximum total rewards for all sensing tasks is

$$\sum_{i=1}^{m} \varphi_i(r_0 + \lambda(N-1)) \le \mathcal{B} \tag{9}$$

That is, the maximum total rewards should not exceed the reward budget $\mathcal{B}$. Given the reward budget $\mathcal{B}$, the increasing scale λ and the demand level N, r_0 can be determined as follows.

$$r_0 = \frac{\mathcal{B}}{\sum_{i=1}^{m} \varphi_i} - \lambda(N-1) \tag{10}$$

V. DISTRIBUTED TASK SELECTION MECHANISMS

In this section, we first prove that both the participatory and opportunistic task selection problems are NP-hard, and then propose two different optimal distributed task selection algorithms respectively to help users select tasks while maximizing their total profits at each sensing round.

Theorem 1: The participatory and opportunistic task selection problems are NP-hard.

Proof: We use a graph to model the task selection problem. Let $G = (V, E, W, R)$ denote the traveling graph for mobile user u_i. $V = \{L_{u_i}, L_{t_1}, L_{t_2}, \cdots, L_{t_m}\}$ denotes the set of vertices consisting of the initial location of user u_i and the locations of all sensing tasks. $R = \{r_{u_i}, r_{t_1}, r_{t_2}, \cdots, r_{t_m}\}$ is the set of weights on vertices where $r_{u_i} = 0$ and r_{t_j} is the reward of task t_j at this round. E is the set of edges between any pair of vertices and W is the set of weights on edges where the weight of an edge is the traveling distance between two vertices.

For the participatory task selection problem, given a set of tasks $\mathcal{T}_{u_i}^k$, $\sum_{t_j \in \mathcal{T}_{u_i}^k} r_{t_j}^k$ can be calculated, and $C(\mathcal{T}_{u_i}^k)$ is the cost on the shortest path that starts from L_{u_i} and travels all the vertices in $\mathcal{T}_{u_i}^k$. Note that the shortest path should be a simple path. When $C(\mathcal{T}_{u_i}^k) = 0$, problem in Eq. 1 is converted to the following problem.

$$\begin{aligned} \max \quad & P(\mathcal{T}_{u_i}^k) = \sum_{t_j \in \mathcal{T}_{u_i}^k} r_{t_j}^k \\ \text{s.t.} \quad & \Gamma_{\mathcal{T}_{u_i}^k} \le B_{u_i}^k \end{aligned} \tag{11}$$

Given the graph G, the problem in Eq. 11 is to find a path originated at L_{u_i} with total travelling time no more

t_6	t_5	t_4	t_3	t_2	t_1
0	1	1	0	1	0

Fig. 3. An example of sequence $\mathbb{i}$ in $dp[\mathbb{i}][j]$ with a total of 6 tasks.

than $B_{u_i}^k$ such that the total rewards gained from vertices is maximized. Hence we can see the problem in Eq. 11 is actually an orienteering problem [32] which is already proved to be NP-hard. Since problem in Eq. 11 is a special case of problem in Eq. 1 where $C(\mathcal{T}_{u_i}^k) = 0$, the participatory task selection problem shown in Eq. 1 is also NP-hard.

For the opportunistic task selection problem, the proof is similar to that of the participatory scenario. The difference is only that the opportunistic task selection asks the user to arrive at the destination. We can also prove that the opportunistic task selection problem is NP-hard similarly. □

In the following, we propose an optimal dynamic programming based algorithm for participatory task selection, and an optimal backtracking based algorithm for opportunistic task selection.

A. Dynamic Programming Based Algorithm for Participatory Task Selection

Given a set of tasks, the total reward is fixed, but the traveling distance is quite different depending on traveling order on the location-dependent tasks. Let $dp[\mathbb{i}][j]$ denotes the shortest path for traveling the set of tasks in $\mathbb{i}$ starting from the initial location of the mobile user and ending at a location L_{t_j} associated with task t_j. Let $dp[\mathbb{i}]$ denotes the shortest path for $\mathbb{i}$, so we can have $dp[\mathbb{i}] = \min_{j=1}^{m}(dp[\mathbb{i}][j])$.

Here $\mathbb{i}$ is a sequence composed of 0 and 1 with the length of m which is the total number of tasks. Thus, $\mathbb{i}$ in the $dp[\mathbb{i}][j]$ ranges from $\{00\cdots0\}$ to $\{11\cdots1\}$. If task t_q is selected by the mobile user, the qth position in sequence $\mathbb{i}$ is 1; otherwise, it is 0. Figure 3 gives an example of sequence $\mathbb{i}$ of $dp[\mathbb{i}][j]$ with a total of 6 tasks. We can see that 1 appears at the second, fourth, and fifth position of the sequence, which means that the tasks t_2, t_4, and t_5 are selected by the mobile user.

Let $dist[j][q]$ denote the distance between task t_j and t_q. Given a sequence of $\mathbb{i}$, we can know the set of tasks selected by the mobile user. Let $o(\mathbb{i})$ denote the performing order of the selected tasks in $\mathbb{i}$. Let $dp[\mathbb{i}][j]_{o(\mathbb{i})}$ denote the total traveling distance starting from the initial location of the mobile user and ending at location of task t_j by following the performing order of $o(\mathbb{i})$. For example, given the sequence in Figure 3, and $o[\mathbb{i}]$ is $\{t_4, t_5, t_2\}$, we have $dp[\mathbb{i}][j]_{o(\mathbb{i})} = dist[s][t_4] + dist[t_4][t_5] + dist[t_5][t_2]$ where s denote the initial location of the mobile user. Obviously, $dp[\mathbb{i}][j]$ should be the shortest path among all the possible traveling paths ending at L_{t_j} for the selected tasks in the sequence $\mathbb{i}$. Note that if task t_j does not belong to the selected tasks in the sequence $\mathbb{i}$, $dp[\mathbb{i}][j]$ should be ∞. Therefore, we have

$$dp[\mathbb{i}][j] = \begin{cases} \min_{o(\mathbb{i})}\{dp[\mathbb{i}][j]_{o(\mathbb{i})}\} & t_j \in \mathbb{i}, \\ \infty & t_j \notin \mathbb{i}. \end{cases} \tag{12}$$

where $t_j \in \mathbb{i}$ means that the jth position of $\mathbb{i}$ is 1.

dp[i] [j]	0	1	2	3	4	5	6
000000	0	∞	∞	∞	∞	∞	∞
000001	∞	10.77	∞	∞	∞	∞	∞
000010	∞	∞	8.60	∞	∞	∞	∞
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
011110	∞	∞	15.41	13.18	17.02	16.14	∞
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
111110	∞	∞	17.59	15.35	19.34	18.32	17.55
111111	∞	18.35	22.82	20.58	24.56	23.54	22.78

Fig. 4. The shortest path matrix of $dp[\mathbb{i}][j]$ with a total of 6 tasks.

For the sequence of $\mathbb{i}$, if we further select another task t_q, then the sequence of $\mathbb{i}$ becomes $\mathbb{i} \mid 1 \ll (q-1)$. $1 \ll (q-1)$ means that 1 shifts to the left by $q-1$ bits, and $\mathbb{i} \mid 1 \ll (q-1)$ means that we take the *or* operation between the sequences of $\mathbb{i}$ and $1 \ll (k-1)$. Thus, we can get a new sequence where t_q is selected besides the previous selected tasks in $\mathbb{i}$. According to Eq. 12, we can have

$$dp[\mathbb{i} \mid 1 \ll (q-1)][q] = \min_{1 \le j \le m} \{dp[\mathbb{i}][j] + dist[j][q]\} \tag{13}$$

From Eq. 13, we can see that finding the shortest path for a set of tasks exhibits optimal substructure, which implies that we can solve the task selection problem with dynamic programming. Therefore, we propose a dynamic programming based task selection algorithm to choose the optimal set of tasks with the maximum profit while satisfying the travel time/distance budget.

The key idea of the algorithm is using a sequence to indicate which task has been selected. The procedures are descried as follows:

1) Construct the shortest path matrix $DP = (dp[\mathbb{i}][j])_{2^m \times (m+1)}$ where m is the number of tasks. $\mathbb{i}$ ranges from $[00\cdots0]$ to $[11\cdots1]$ and j ranges from 0 to m. $dp[00\cdots0][0]$ is initialized with 0 and all the other entries are initialized with ∞.
2) Calculate all $dp[\mathbb{i}][j]$ according to Eq. 13.
3) Calculate the total profits for each $\mathbb{i}$, denoted by $P(\mathbb{i}) = R(\mathbb{i}) - C(\mathbb{i})$, where $R(\mathbb{i})$ is the total rewards of selected tasks in sequence $\mathbb{i}$, and $C(\mathbb{i})$ is the traveling cost corresponding to the shortest path $dp[\mathbb{i}]$.
4) Find the maximum $P(\mathbb{i})$ whose shortest path $dp[\mathbb{i}]$ is no larger than the traveling time/distance budget.

Figure 4 shows the shortest path matrix of $dp[\mathbb{i}][j]$ with a total of 6 tasks. $\mathbb{i}$ ranges from $\{000000\}$ to $\{111111\}$ and j ranges from 0 to 6. $dp[000000][0]$ is set to 0 while other entries are set to ∞. We calculate $dp[\mathbb{i}][j]$ one by one according to Eq. 13, so the shortest path for $\mathbb{i}$, $dp[\mathbb{i}]$, can be easily obtained. For each row of sequence $\mathbb{i}$, the total rewards $R(\mathbb{i})$ can be easily calculated by summing up the rewards of selected tasks in $\mathbb{i}$. Finally, we filter out the sequences whose shortest path does not meet the traveling time/distance budget and find out the maximum $P(\mathbb{i})$ from the remaining sequences. Thus, the selected tasks in the corresponding sequence $\mathbb{i}$ is the optimal set of tasks for the mobile user to perform.

Theorem 2: The dynamic programming based task selection algorithm has a computational complexity of $O(m^2 2^m)$, where m is the total number of tasks.

Proof: The shortest path matrix DP has $2^m * (m+1)$ entries, where m is the number of tasks. For calculating each entry $dp[i][j]$, it needs to run m steps according to Eq. 13. Therefore, the computational complexity of the dynamic programming based task selection algorithm is $O(m^2 2^m)$. □

B. Backtracking Based Algorithm for Opportunistic Task Selection

The opportunistic task selection problem is similar with the orienteering problem but has a fixed destination. In this paper, we leverage backtracking to search the optimal solution. The solution space of this problem is a permutation tree, and we need to find an optimal permutation of tasks that could maximize the profit while satisfying budget.

Given a start and end point of a user u_i, we can construct an ellipse over the whole task set, denoted by EP_{u_i} for u_i, by using the start point and end point as the two foci of the ellipse, and the travel distance within the time budget of $B_{u_i}^k$ as the length of the major axis. Then the traveling time to perform any task outside the ellipse will exceed the budget, so we could just consider tasks within the ellipse when searching the optimal path. This will significantly reduce the searching space for the optimal path.

Let T_{u_i} denote the set of tasks inside the ellipse for u_i. For each task t_i in T_{u_i}, the traveling time from the start point through t_i to the end point is definitely less than the time budget $B_{u_i}^k$ of the user, because the distance between any point inside the ellipse and its two foci is smaller than its major axis. Let opt_{u_i} denote the optimal path for u_i, which can be generated by using depth first search (DFS) on the permutation tree of T_{u_i}, and P_{u_i} denote the maximum profits in all the feasible paths.

We extend the branch of each task t_{i_j} in T_{u_i} by using DFS to get a feasible path $f_{t_{i_j}}$. After searching all the branches started with each t_{i_j}, we can get a set of feasible paths, denoted by $\mathcal{F} = \{f_{t_1}, f_{t_2} \cdots f_{t_{|T_{u_i}|}}\}$. Finally, we can find the optimal path opt_{u_i} that can maximize profit for user u_i from the feasible set $\mathcal{F}$.

Specifically, for each task t_{i_j} in T_{u_i}, we first add it to a feasible path $f_{t_{i_j}}$ as the first task in current branch, and the cost for traversing $f_{t_{i_j}}$ is denoted by $C(f_{t_{i_j}})$. Then we keep searching with DFS in that branch and adding tasks in $f_{t_{i_j}}$ until $C(f_{t_i})$ exceeds the budget $B_{u_i}^k$. Then we update the path if the profit of this branch is larger than the records, back to the pervious layer and keep searching others branch.

The pseudocode of the backtracking is formally presented in Algorithm 1.

Theorem 3: The backtracking based path selection algorithm has a computational complexity of $O(\frac{\bar{m}!}{(\bar{m}-\bar{n})!})$, where $\bar{m}$ is the total number of tasks inside of the ellipse, and $\bar{n}$ is the average number of tasks that users select.

Proof: If the budget of users is large enough, all the permutations of $\bar{m}$ tasks will be $\bar{m}!$. However, users typically only have limited budgets. Let us assume the average number of tasks one user can perform is $\bar{n}$ within his budget, which is usually a very small number. Then the average depth of the permutation tree is $\bar{n}$, and any branches that exceed the budget will be pruned. Each mobile user selects the first task from $\bar{m}$ tasks, and then the second from $\bar{n}-1$ tasks, until the time cost exceeds the budget. On average, each user will select about $\bar{n}$ tasks at most. Therefore, the computational complexity of the backtracking based path selection algorithm is $O(\frac{\bar{m}!}{(\bar{m}-\bar{n})!})$. Given that both $\bar{m}$ and $\bar{n}$ are small, the computational cost is affordable. □

Algorithm 1 Backtracking Based Path Selection for a User

Require: Task set T_{u_i}; budget $B_{u_i}^k$
Ensure: The optimal path opt_{u_i}
Initialize path set $\mathcal{F}$
for each $t_{i_j} \in T_{u_i}$ **do**
  add t_{i_j} as the first task in $f_{t_{i_j}}$
  $f_{t_{i_j}}, P_{t_{i_j}} \leftarrow DFS(T_{u_i} \backslash t_{i_j}, f_{t_{i_j}}, C(f_{t_{i_j}}))$
  $\mathcal{F} \leftarrow \mathcal{F} \cup f_{t_{i_j}}$
end for
$opt_{u_i} \leftarrow$ the path has max profit in $\mathcal{F}$
return opt_{u_i}

Algorithm 2 DFS$(T_{u_i}, f_{t_{i_j}}, C(f_{t_{i_j}}))$

for each $t_{i_j} \in T_{u_i}$ **do**
  $f_{t_{i_j}} \leftarrow f_{t_{i_j}} \cup t_{i_j}$
  if $C(f_{t_{i_j}}) < B_{u_i}^k$ **then**
    $\alpha, \beta \leftarrow DFS(T_{u_i} \backslash t_{i_j}, f_{t_{i_j}}, C(f_{t_{i_j}}))$
  else
    if $\beta > P_{t_{i_j}}$ **then**
      $f_{t_{i_j}} \leftarrow \alpha$, $P_{t_{i_j}} \leftarrow \beta$
    end if
    back to the pervious layer
  end if
end for
return $f_{t_{i_j}}, P_{t_{i_j}}$

VI. Performance Evaluation

In this section, we evaluate the performance of the proposed demand-driven dynamic incentive mechanism under three different task selection scenarios, participatory, opportunistic and hybrid scenarios. Then we compare our algorithm in hybrid scenario with the steered crowdsensing mechanism [11] which dynamically changes the rewards of tasks according to expected quality improvements. As for the dynamic incentive mechanisms proposed in [9] and [13], the former is proposed to maintain adequate level of participants and does not take location into consideration, while the latter is designed for one-shot sensing and the budget of each task would not change once initially determined, so they are not suitable to compare with our mechanism. Moreover, we also compare our incentive mechanism with a fixed incentive mechanism where the reward would not change once determined.

Steered crowdsensing mechanism: In [11], the reward of a task changes dynamically according to the expected quality

TABLE IV
PARAMETER SETTING DESCRIPTION

Parameter	value
Tasks Number m	20
Independent Measurements per Task φ_i	20
Mobile Users Number n	[40, 140]
Initialized Reward r_0	0.5\$
Demand Increasing Scale λ	0.5\$
Total Reward Budget $\mathcal{B}$	1000\$
Demand Factor Coefficient λ_1, λ_2, λ_3	100

improvements of the task. The reward function (Eq. 12 in [11]) of the steered crowdsensing mechanism is rewritten as follows.

$$R_{t_i}^k = R_c + \mu \Delta Q(x) \tag{14}$$

where $R_{t_i}^k$ is the reward of task t_i at the kth round, R_c is an additional reward given to the participant, and $\Delta Q(x) = Q(x+1) - Q(x)$ is the expected quality improvement due to received $(x+1)$th measurement of the task. In our experiments, we set $\mu = 100$, $\delta = 0.2$, $r_c = 5$, so the reward of each task varies in [5, 25].

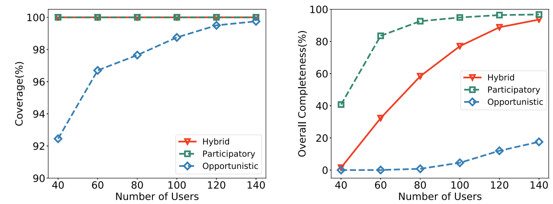
It is worth noting that the reward function of the steered crowdsensing incentive mechanism in Eq. 14 looks similarly to our demand-driven dynamic reward function in Eq. 8. However, the reward function of steered incentive is a decreasing function which becomes smaller and smaller as more measurements are received. In this way, the attraction of each task to participants becomes smaller and smaller as time goes on. In contrast, our demand-driven function is determined by the demand of each task but not the expected quality improvement, so it can increase when demand is high and also can decrease when the demand is small.

Fixed incentive mechanism: The fixed incentive mechanisms set a fixed reward for each task and the reward would not change once it is initially determined. In our experiments, we also compare the proposed demand-driven dynamic incentive mechanism with the fixed incentive mechanism. In each experiment, the fixed incentive mechanism randomly generates a demand level for each task as presented in Table III and uses the corresponding reward for each task. The reward of each task would not change in latter rounds.

In our experiments, the locations of mobile users and sensing tasks are randomly generated in a 3000m $\times$ 3000m area. We assume each mobile user's walking speed is $2m/s$ and the cost for movement is $0.002\$/m$. We assume there are 20 sensing tasks and each sensing task requires 20 independent measurements to reach the required quality. The deadline of each sensing task is randomly generated between $[5, 15]$. Given the reward budget $\mathcal{B} = 1000\$$, we map the demand into five demand levels as shown in Table III and set $\lambda = 0.5\$$ and $r_0 = 0.5\$$. The number of mobile users ranges from 40 to 140. We perform each experiment for 100 times and use the average value to demonstrate the performance. The parameter setting description is shown in Table IV.

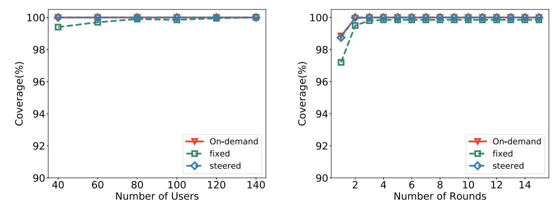
A. Comparison of Scenarios

We first compare the coverage and the overall completeness of sensing tasks under the three different scenarios. The coverage refers to how many tasks have been selected, and



(a) (b)

Fig. 5. The comparison of coverage and overall completeness under three different scenarios.



(a) (b)

Fig. 6. The comparison of the incentive mechanisms on the coverage.

the overall completeness refers to how many tasks have gotten enough measurements.

Figure 5(a) show the coverage of the three scenarios against the number of users until the last sensing round. The participatory and hybrid scenarios always achieve 100% coverage, while the coverage of the opportunistic scenario cannot reach 100% even for 140 mobile users. This is because many users in the participatory and hybrid scenarios have more chance to select far away tasks as they do not have requirements of reaching some destination at final. However, the users in the opportunistic scenario cannot select far away tasks from their daily routes.

Figure 5(b) shows the overall completeness of three scenarios. We can see that the overall completeness of the participatory scenario is always higher than the other two scenarios. The opportunistic scenario shows the worst performance and the overall completeness is only 20% even there are 140 mobile users. It is worth noting that the hybrid scenario behaves better when more users perform sensing tasks and achieves similar performance when there are 140 mobile users. This implies that we can recruit some opportunistic users to improve system performance when not enough participatory users can be recruited.

In the following experiments, we consider the hybrid scenario with both participatory and opportunistic users, and compare the proposed demand-driven incentive mechanism with the steered and fixed mechanisms.

B. Coverage

Coverage measures how good the algorithm balances the popularity among sensing tasks, which is a kind of spatial metric. The larger the coverage, the better the balance.

Impact of user number: Figure 6(a) shows the coverage of the three mechanisms against the number of users until the last sensing round. We can see that the demand-driven incentive mechanism and the steered crowdsensing incentive mechanism always achieve better coverage than the fixed incentive mechanism. The demand-driven incentive mechanism and the steered crowdsensing incentive mechanism always achieve 100% coverage which means that each sensing task is at least selected once by users. This is because our algorithm can characterize the demand of each task from multiple factors and change the relative popularity among tasks, so that even far away sensing tasks will be selected by mobile users. As for the steered crowdsensing incentive mechanism, the rewards of sensing tasks without receiving any measurement become relatively higher compared to others, which encourages mobile users to select these uncovered sensing tasks. While the coverage for the fixed incentive mechanism increases as the increasing of the number of mobile users, since more users means higher probability of a task to be selected/covered. However, the fixed incentive mechanism cannot reach 100% coverage even for 140 mobile users.

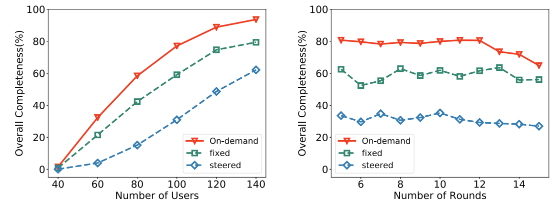
Impact of sensing rounds: Figure 6(b) shows the coverage of the three mechanisms against the number of sensing rounds when there are 100 mobile users. First, we can observe that the coverage of the demand-driven incentive mechanism and the steered crowdsensing incentive mechanism are always higher than that of the fixed incentive mechanism at all sensing rounds. We can also see that the coverage increases at first as the round goes on since more uncovered tasks will be selected. The coverage of the demand-driven incentive mechanism and the steered incentive mechanism reaches 100% coverage while that of the fixed incentive mechanism cannot reach 100% coverage. This means that just increasing the sensing rounds does not increase the popularity of unpopular sensing tasks in the fixed incentive mechanism.

C. Overall Completeness

Each sensing task is expected to be completed before its deadline and the overall completeness measures how good of task completeness before their deadlines.

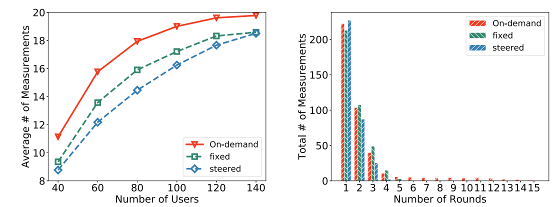
Impact of user number: Figure 7(a) shows the overall completeness of the three mechanisms against the number of users until the last sensing round. The overall completeness grows as the number of mobile users increases. We can see that the demand-driven incentive mechanism has a higher overall completeness than the fixed and the steered incentive mechanism, and the superiority becomes more obvious when there are more mobile users. Compared to Figure 5(b), the overall completeness of all the three mechanisms is lower than the completeness of participatory scenario as half of mobile users in hybrid scenario have less opportunity to select tasks far away from their destinations.

Impact of sensing rounds: Figure 7(b) shows the overall completeness of the three mechanisms against the number of sensing rounds when there are 100 mobile users. The deadline of each sensing task is randomly generated between $[5, 15]$. We can see that the demand-driven incentive mechanism always has a higher overall completeness than the fixed incentive mechanism and the steered incentive mechanism for all sensing rounds. The demand-driven incentive mechanism achieves about 80% completeness while the fixed incentive mechanism only has about 60% completeness. The steered crowdsensing



(a) (b)

Fig. 7. The comparison of the incentive mechanisms on the overall completeness.



(a) (b)

Fig. 8. The comparison of the incentive mechanisms on the # of measurements.

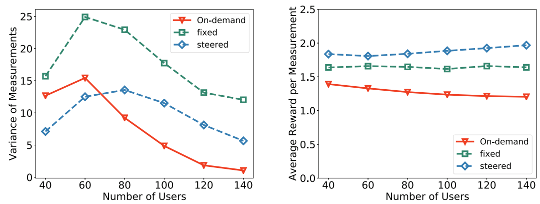
incentive mechanism has the worst performance that only achieves 30% completeness since it changes reward only based on the quality of sensing tasks and steers users to select sensing tasks with lower quality but does not take the deadline of tasks into consideration. As higher measurement tasks have less popularity, which result in the lower completeness of steered incentive mechanism.

D. # of Measurements

Each sensing task expects to receive the required number of measurements before its deadline to ensure the sensing quality. In particular, the more number of measurements, the better encouragement given by the incentive mechanisms.

Impact of user number: Figure 8(a) shows the comparison of the incentive mechanisms on the average # of measurements of all sensing tasks against the number of users until the last sensing round. In our experiments, 20 measurements are required for each sensing task. The average # of measurement increases as the number of mobile users increases, as there are more users to work on the tasks. We can observe that the on-demand incentive mechanism achieves the best performance compared to the other incentive mechanisms and its average # of measurements can reach almost 20 when there 120 mobile users.

Impact of sensing rounds: Figure 8(b) shows the total # of measurements of all tasks at a round when there are 100 mobile users. As shown in Figure 8(b), the steered incentive mechanism has the largest total number of measurements at the first round, which is because its rewards are higher than the others at this round given the reward update rule in Eq. 14. The fixed incentive mechanism performs better at the following 2nd and 3rd round than the on-demand and steered incentive mechanisms. This is because the rewards of the on-demand and the steered incentive mechanisms decrease as tasks receive more and more measurements, while the the rewards of fixed incentive mechanism do not change and are relatively higher than that of the other two incentive mechanisms. Starting from



(a) (b)

Fig. 9. The comparison of the incentive mechanisms on variance of measurements and average reward per measurement.

the 6th round, there is no more new measurement for the fixed and the steered incentive mechanisms, which is because the rewards cannot encourage mobile users to perform far-away tasks. In contrast, the proposed on-demand incentive mechanism continually has new measurements for the tasks at the following rounds, which is because that it dynamically change the rewards of tasks according to their real-time demands, which can encourage users to perform far-away tasks.

E. Variance of Measurements

The variance of measurements characterizes the balance of users' participation among sensing tasks. If an incentive mechanism achieves larger average # of measurements with smaller variance of measurements than others, it achieves better balance of users' participation among sensing tasks.

Figure 9(a) shows the variance of measurement of the three mechanisms against the number of the users until the last sensing round. We can first observe that the variance of measurements of the on-demand incentive mechanism is smaller than the other two incentive mechanisms when the total number of users is more than 60. Given that it also has the largest average # of measurements as shown in Figure 8(a), we can conclude that the proposed on-demand incentive mechanism realizes better balance of users' participation among sensing tasks.

Note that the variance of measurements of the three incentive mechanisms tends to decrease with more users. This is because users tend to select nearby sensing tasks and more users means better distribution of measurements among tasks. The variance of measurements steered incentive mechanism is lower than on-demand incentive mechanism when the number of users is less than 60, this is because the overall completeness of tasks is relatively low as shown in Figure 7(a), it is the lacking of enough users that affect the balance efficiency of of multi-factor demand based incentive mechanism.

F. Average Reward per Measurement

The platform always expects to maximize its welfare and we use the reward per measurement to reflect this objective. The platform will have a larger welfare if it pays smaller reward per measurement.

Figure 9(b) shows the average reward per measurement of the three mechanisms against the number of users until the last sensing round. We can see the average reward per measurement of the on-demand incentive mechanism is smaller than that of the fixed incentive mechanism and the steered incentive mechanism. This is because our algorithm can find more suitable values for the rewards according to the demands of tasks while the rewards of sensing tasks in the fixed incentive mechanism cannot change. The average reward per measurement of the on-demand incentive mechanism decreases as the increasing of the mobile users, since the demand is stronger for less number of mobile users.

VII. Conclusion

In this paper, we focused on location-dependent crowdsensing systems, and proposed a demand-driven dynamic incentive mechanism that dynamically changes the reward of each task in an on-demand way to balance the popularity among tasks. We introduced the demand indicator which uses the deadline, the completing progress, and the number of neighboring mobile users to characterize the real-time demand of each sensing task. At each sensing round, we used the framework of AHP to calculate the relative demands of all sensing tasks and then determine their rewards. Moreover, we considered the task selection problem under the participatory scenario and the opportunistic scenario, and proposed the dynamic programming based solution and the backtracking based solution, respectively. Extensive experiments show that the proposed dynamic incentive mechanism outperforms the state-of-the-art in terms of coverage, overall completeness, and the average reward per measurement. That is, the proposed demand-driven dynamic incentive mechanism achieves better participation and participation balance among tasks.

References

[1] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw. (Mobicom)*, 2012, pp. 173–184.

[2] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," in *Proc. ACM SenSys*, 2008, pp. 323–336.

[3] P. Dutta *et al.*, "Common sense: Participatory urban sensing using a network of handheld air quality monitors," in *Proc. 7th ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, 2009, pp. 349–350.

[4] I. Schweizer, C. Meurisch, J. Gedeon, R. Bärtl, and M. Mühlhäuser, "Noisemap: Multi-tier incentive mechanisms for participative urban sensing," in *Proc. 3rd Int. Workshop Sens. Appl. Mobile Phones (PhoneSense)*, 2012, pp. 1–5.

[5] T. Liu and Y. Zhu, "Social welfare maximization in participatory smartphone sensing," *Comput. Netw.*, vol. 73, pp. 195–209, Nov. 2014.

[6] J. Sun and H. Ma, "Heterogeneous-belief based incentive schemes for crowd sensing in mobile social networks," *J. Netw. Comput. Appl.*, vol. 42, pp. 189–196, Jun. 2014.

[7] Y. Zhang and M. van der Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2140–2148.

[8] D. Zhao, X.-Y. Li, and H. Ma, "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2014, pp. 1213–1221.

[9] J.-S. Lee and B. Hoh, "Dynamic pricing incentive for participatory sensing," *Pervas. Mobile Comput.*, vol. 6, no. 6, pp. 693–708, Dec. 2010.

[10] H. Jin, L. Su, D. Chen, K. Nahrstedt, and J. Xu, "Quality of information aware incentive mechanisms for mobile crowd sensing systems," in *Proc. 16th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2015, pp. 167–176.

[11] R. Kawajiri, M. Shimosaka, and H. Kashima, "Steered crowdsensing: Incentive design towards quality-oriented place-centric crowdsensing," in *Proc. ACM UbiComp*, 2014, pp. 691–701.

[12] D. Peng, F. Wu, and G. Chen, "Pay as how well you do: A quality based incentive mechanism for crowdsensing," in *Proc. 16th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2015, pp. 177–186.

[13] B. Guo *et al.*, “TaskMe: Toward a dynamic and quality-enhanced incentive mechanism for mobile crowd sensing,” *Int. J. Hum.-Comput. Stud.*, vol. 102, pp. 14–26, Jun. 2017.

[14] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, “TRAC: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing,” in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2014, pp. 1231–1239.

[15] L. Gao, F. Hou, and J. Huang, “Providing long-term participation incentive in participatory sensing,” in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 2803–2811.

[16] S. He, D.-H. Shin, J. Zhang, and J. Chen, “Toward optimal allocation of location dependent tasks in crowdsensing,” in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2014, pp. 745–753.

[17] Z. Wang *et al.*, “Heterogeneous incentive mechanism for time-sensitive and location-dependent crowdsensing networks with random arrivals,” *Comput. Netw.*, vol. 131, pp. 96–109, Feb. 2018.

[18] Z. Wang, J. Hu, J. Zhao, D. Yang, H. Chen, and Q. Wang, “Pay on-demand: Dynamic incentive and task selection for location-dependent mobile crowdsensing systems,” in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 611–621.

[19] H. To, C. Shahabi, and L. Kazemi, “A server-assigned spatial crowdsourcing framework,” *ACM Trans. Spatial Algorithms Syst.*, vol. 1, no. 1, pp. 1–28, Aug. 2015.

[20] J.-S. Lee and B. Hoh, “Sell your experiences: A market mechanism based incentive for participatory sensing,” in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. (PerCom)*, Mar. 2010, pp. 60–68.

[21] D. Yang, X. Fang, and G. Xue, “Truthful incentive mechanisms for k-anonymity location privacy,” in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 2994–3002.

[22] I. Krontiris and A. Albers, “Monetary incentives in participatory sensing using multi-attributive auctions,” *Int. J. Parallel, Emergent Distrib. Syst.*, vol. 27, no. 4, pp. 317–336, Aug. 2012.

[23] T. Luo, H.-P. Tan, and L. Xia, “Profit-maximizing incentive for participatory sensing,” in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2014, pp. 127–135.

[24] M. H. Cheung, R. Southwell, F. Hou, and J. Huang, “Distributed time-sensitive task selection in mobile crowdsensing,” in *Proc. ACM MobiHoc*, 2015, pp. 157–166.

[25] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, “Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy,” *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 4, pp. 591–606, Jul./Aug. 2018.

[26] Z. Wang *et al.*, “When mobile crowdsensing meets privacy,” *IEEE Commun. Mag.*, vol. 57, no. 9, pp. 72–78, Sep. 2019.

[27] L. Zhao, Q. Wang, Q. Zou, Y. Zhang, and Y. Chen, “Privacy-preserving collaborative deep learning with unreliable participants,” *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1486–1500, 2020, doi: 10.1109/TIFS.2019.2939713.

[28] K. Ren, Q. Wang, C. Wang, Z. Qin, and X. Lin, “The security of autonomous driving: Threats, defenses, and future directions,” *Proc. IEEE*, vol. 108, no. 2, pp. 357–372, Feb. 2020.

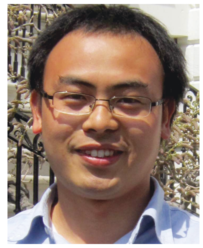
[29] Y. Zheng, H. Duan, and C. Wang, “Learning the truth privately and confidently: Encrypted confidence-aware truth discovery in mobile crowdsensing,” *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 10, pp. 2475–2489, Oct. 2018.

[30] Z. Wang *et al.*, “Personalized privacy-preserving task allocation for mobile crowdsensing,” *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1330–1341, Jun. 2019.

[31] T. Saaty, *The Analytic Hierarchy Process*. New York, NY, USA: McGraw-Hill, 1980.

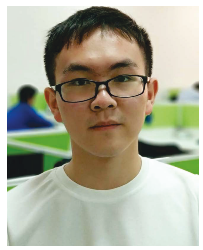
[32] A. Blum, S. Chawla, D. R. Karger, T. Lane, A. Meyerson, and M. Minkoff, “Approximation algorithms for orienteering and discounted-reward TSP,” *SIAM J. Comput.*, vol. 37, no. 2, pp. 653–670, Jan. 2007.

Jiahui Hu (Member, IEEE) received the B.S. degree in information security in 2016 and the master’s degree from the School of Cyber Science and Engineering, Wuhan University, in 2019. Her research interest focuses on mobile crowdsensing systems.

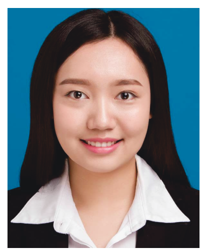


Zhibo Wang (Senior Member, IEEE) received the B.E. degree in automation from Zhejiang University, China, in 2007, and the Ph.D. degree in electrical engineering and computer science from The University of Tennessee, Knoxville, in 2014. He is currently a Professor with the School of Cyber Science and Engineering, Wuhan University, China. His current research interests include mobile crowdsensing and the Internet of Things. He is a member of the ACM.

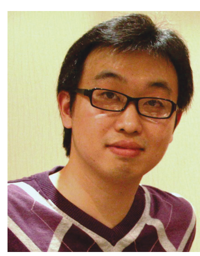
Jian Wei received the B.S. degree in information and computing science from Wuhan Textile University, China, in 2017, and the master’s degree from the School of Cyber Science and Engineering, Wuhan University, in 2019. His research interest focuses on mobile crowdsensing systems.



Ruizhao Lv received the B.S. degree in computing science and technology from Northeast Agricultural University, China, in 2017, and the master’s degree from the School of Cyber Science and Engineering, Wuhan University, in 2019. His research interest focuses on mobile crowdsensing systems.

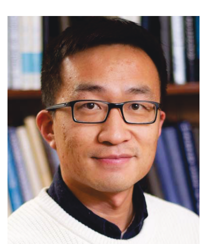


Jing Zhao received the B.S. degree in information security from Wuhan University, China, in 2017. She is currently pursuing the Ph.D. degree with Peking University, China. Her research interests include mobile crowdsensing systems and image processing.



Qian Wang (Senior Member, IEEE) received the B.S. degree from Wuhan University, China, in 2003, the M.S. degree from the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, China, in 2006, and the Ph.D. degree from the Illinois Institute of Technology, USA, in 2012, all in electrical engineering. He is currently a Professor with the School of Cyber Science and Engineering, Wuhan University. His research interests include wireless network security and cloud computing security.

Honglong Chen (Member, IEEE) received the B.E. degree in automation from the China University of Petroleum, China, in 2006, the M.E. degree from the Department of Control, Zhejiang University, China, in 2008, and the Ph.D. degree in computer science from The Hong Kong Polytechnic University, Hong Kong, in 2012. He is currently an Associate Professor with the College of Information and Control, China University of Petroleum, China. His research interests are in the areas of delay tolerant networks, and security and privacy.



Dejun Yang (Senior Member, IEEE) received the B.S. degree from Peking University, Beijing, China, in 2007, and the Ph.D. degree in computer science from Arizona State University, Tempe, AZ, USA, in 2013. He is currently the Ben L. Fryrear Assistant Professor of computer science with the Colorado School of Mines, Golden, CO, USA. His research interests include economic and optimization approaches to networks, crowdsourcing, smart grid, and security and privacy.