Pay On-demand: Dynamic Incentive and Task Selection for Location-dependent Mobile Crowdsensing Systems

Zhibo Wang†, Jiahui Hu†, Jing Zhao†, Dejun Yang‡, Honglong Chen§, Qian Wang†
†School of Computer Science, Wuhan University, P. R. China
‡Deptment of Computer Science, Colorado School of Mines, USA
§College of Information and Control Science, China University of Petroleum, P. R. China

***Abstract*—With the rich sensing capacity and ubiquitous usage of smartphones, crowdsensing leveraging the power of the crowd of mobile users has become an effective technique to collect data for various sensing applications. Many incentive mechanisms have been proposed to encourage people to participate in crowdsensing. 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. In this paper, we focus on location-dependent crowdsensing systems and propose a demand-based 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 to calculate the relative demands of all sensing tasks and then determine their rewards accordingly. Moreover, we prove that the distributed task selection problem with time budget is NP-hard. We propose an optimal dynamic programming based solution and a greedy solution to help each user select tasks while maximizing its profit. Extensive experiments show that the demand-based dynamic incentive mechanism outperforms existing incentive mechanisms.**

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, accelerator, GPS, and compass). 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 mobile crowdsensing 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.

Recently several location-dependent sensing applications have adopted crowdsensing to collect massive data, such as air quality monitoring [2], wifi signal map construction [3], traffic condition monitoring [4], and noise pollution assessment [5], [6]. The sensing tasks are location dependent where each task should be performed at a specific location and requires mobile users to contribute the location related sensing data to complete the task. Since the quality of sensing data varies from person to person, multiple users are expected to contribute their sensing data to improve the quality of completing a task.

When mobile users contribute sensing data in crowdsensing, they spend not only time but also physical resources to complete sensing tasks. Without an appropriate incentive, mobile users may not be willing to participate in crowdsensing. Moreover, the privacy leakage concern from privacy sensitive mobile users further prevents users from contributing sensing data. Recently many work have been focused on the incentive mechanism design to improve the willingness of mobile users to participate in crowdsensing. Some of them are game-theoretic incentive mechanisms that allocate tasks to mobile users with the objective of maximizing the social surplus [7]–[11]. Some of them are quality-orientated incentive mechanisms designed to improve the quality of sensing data [12]–[15]. Moreover, with the rapid development of mobile devices, incentive mechanisms for location-dependent crowdsensing systems are proposed in [13], [15]–[19].

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 import factor besides the reward influencing the decision of users to perform a sensing task or not. In this paper, we argue that there exists inherent inequality among location-dependent sensing tasks and the demands of tasks for participants dynamically change as time goes on. In general, mobile users prefer to perform close tasks with high rewards, while far away tasks with low rewards will be ignored. That is, the location difference results in popularity difference of tasks to participants. Hence, the popularity of a task is inherently determined from the beginning by its location and its initial reward in fixed incentive mechanisms, which leads to a low coverage issue that only popular tasks can be completed while unpopular location-dependent sensing tasks 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 some dynamic incentive mechanisms [11] [13] [15]are also proposed. In [11], the authors proposed a reverse auction-based dynamic pricing incentive mechanism for participatory sensing to maintain adequate level of participants, which however does not take location difference and demand difference into consideration. Guo et al. proposed a quality-oriented dynamic incentive mechanism that sets different/dynamic budget value for each sensing task according to the spatio-temporal popularity level [15]. However, the proposed incentive mechanism focuses on one-shot sensing tasks. Although the budget of each task is different from another, it would not change once initially determined. Therefore, it can be considered as a fixed incentive mechanism with different budgets for each task. Kawajiri et al. [13] proposed a steered incentive mechanism which changes points (rewards) in every session to improve the quality of service rather than data size. However, the point decreases as the time goes on, which results in less and less engagement of participants. Moreover, this paper did not take the deadline difference of sensing tasks into consideration.

In this paper, we focus on location-dependent crowdsensing 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 way, 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-based dynamic incentive mechanism and distributed task selection algorithms 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, the Analytic Hierarchy Process is adopted to model and calculate the relative demands of all sensing tasks and then their rewards can be determined accordingly.

The main contributions of this paper are summarized as follows.

- We propose a demand-based dynamic incentive mechanism for location-dependent crowdsensing 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 distributed task selection problem is NP-hard. To solve this problem, we first propose an optimal dynamic programming solution, and further propose an efficiently greedy solution to help mobile users select appropriate set of tasks while maximizing their profits at each sensing round.
- We conduct extensive experiments to compare the proposed demand-based dynamic incentive mechanism with existing incentive mechanisms. The experimental results show that the proposed mechanism achieves better participation and participation balance among tasks.

The remainder of this paper is organized as follows. We present the system overview and describe the task selection and incentive design problems in Section III. We present the demand-based dynamic incentive mechanism in Section IV, and the 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

Location-dependent incentive mechanisms have received great attention in recent years. In [20], the authors classified the location-dependent crowdsensing problem into two modes: Worker Selected Tasks (WST) and Server Assigned Tasks (SAT).

For the SAT mode, the server has the global information of the tasks as well as mobile users and the auction-based mechanisms are usually used to assign tasks to mobile users. Lee et al. [21] applied reverse auctions in the economic field to the research of crowdsensing incentive mechanisms, which ensured the relatively high participation of users while minimizing the payment cost. Feng et al. [17] designed a truthful mechanism that used reverse combinatorial auction model to motivate the participants. Krontiris et al. [22] used the multi-attribute auction mechanism in reverse auction, which not only considered the participation rate of users, but also took account of the quality of sensing data. In [23], a full-pay auction method was proposed to motivate participants to participate in, of which only the bidder that contributes mostly can get the payoff. Yang et al. [24] used double auction mechanism to motivate participants to join the K anonymity of location-sensitive. In [18], the VCG auction mechanism was adopted and the updating rule was introduced aiming at

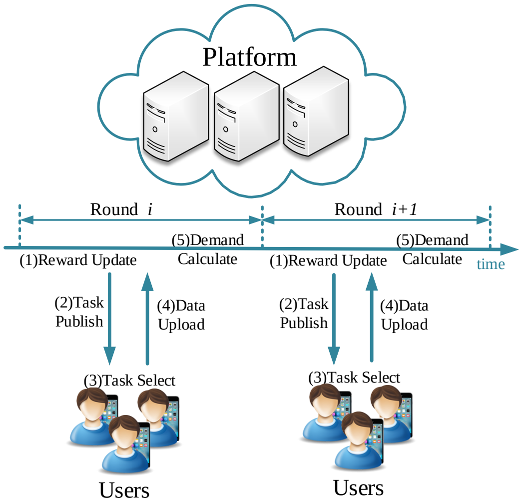


Fig. 1. The architecture of crowdsensing systems with the dynamic incentive mechanism

the online crowdsensing incentive mechanism, the objection of which was to maximize the social welfare benefits.

In the WST mode, mobile users can select any tasks without contacting with the server. Although it is difficult to achieve the objective of maximization in the WST mode, it's actually the typical mode in some popular crowdsensing system such as Gigwalk, Amazon Mechanical Turk, and Field Agent. Moreover, Kawajiri et al. [13] proposed steered crowdsensing, which controls the incentives of users by using the game elements on location-based services. In [25], the authors proposed an asynchronous and distributed task selection algorithm to help workers find a best schedule. Mobile users can submit less personal information compared to the SAT mode, which can improve the participation of mobile users. Furthermore, the procedure of the WST mode is more concise than the SAT mode. However, one drawback of this mode is that the server does not have any control over the allocation of sensing tasks. This may result that some sensing tasks cannot be completed, while others are completed redundantly.

In this paper, we focus on the WST mode and propose a demand-based dynamic incentive mechanism that dynamically changes the rewards of sensing tasks in an on-demand way for location-dependent crowdsensing systems.

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 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-based dynamic incentive mechanism for location-dependent crowdsensing 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.

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 mobile crowdsensing. 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. Generally speaking, 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.

Since sensing tasks are location-dependent, a mobile user has to travel multiple locations to perform multiple sensing tasks. At each sensing round, a mobile user is assumed to have a time budget to perform tasks. 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 task selection problem for the mobile user u_i can be formulated as follows:

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

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 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.

IV. Demand-based Dynamic Incentive Mechanism

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 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.

$$d^k_i = w_1 X^k_{i_1} + w_2 X^k_{i_2} + w_3 X^k_{i_3} \tag{2}$$

where $X^k_{i_1}$, $X^k_{i_2}$ and $X^k_{i_3}$ 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) [26] 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

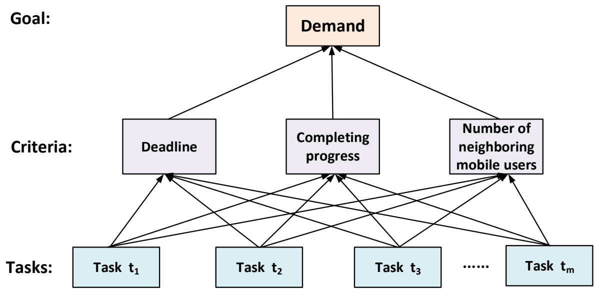


Fig. 2. The hierarchical structure of deciding the demands of tasks

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:

$$X_{i_1}^k = \lambda_1 \ln(1 + \frac{1}{\tau_i - (k-1)}) \tag{3}$$

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

$$X_{i_2}^k = \lambda_2 \ln(1 + (1 - \frac{\pi_i}{\varphi_i})) \tag{4}$$

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

$$X_{i_3}^k = \lambda_3 \ln(1 + (1 - \frac{N_i}{N_{max}})) \tag{5}$$

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. 2 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.

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

TABLE I
AN EXAMPLE OF PAIRWISE COMPARISON MATRIX $A = (a_{ij})_{3\times3}$

Pairwise Comparison Matrix A: We use the pairwise comparison matrix $A = (a_{ij})_{3\times3}$ 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 $A = (a_{ij})_{3\times3}$. Each entry a_{ij} represents the relative importance of the criteria 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 [26]. 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 $\bar{A} = (\bar{a}_{ij})_{3\times3}$ 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.

	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

TABLE II
NORMALIZED PAIRWISE COMPARISON MATRIX $\bar{A} = (\bar{a}_{ij})_{3\times3}$ FOR THE EXAMPLE IN TABLE I

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,

$$w_i = \frac{1}{3}\sum_{j=1}^{3} \bar{a}_{ij} \tag{6}$$

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 $0 \leq X_{i_1}^k \leq \lambda_1 \ln 2$, $0 \leq X_{i_2}^k \leq \lambda_2 \ln 2$ and $0 \leq X_{i_3}^k \leq \lambda_3 \ln 2$, and $w_1 + w_2 + w_3 = 1$, we can have $d_i^k = w_1 X_{i_1}^k + w_2 X_{i_2}^k + w_3 X_{i_3}^k \leq \lambda_{max} \ln 2$ 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. 2. 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 $0 \leq X_{i_1}^k \leq \lambda_1 \ln 2$, $0 \leq X_{i_2}^k \leq \lambda_2 \ln 2$ and $0 \leq X_{i_3}^k \leq \lambda_3 \ln 2$, and $w_1 + w_2 + w_3 = 1$, we can have $d_i^k \leq \lambda_{max} \ln 2$ 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 $(0.2, 0.4]$.

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

TABLE III
AN EXAMPLE OF DEMAND LEVELS WHEN $N = 5$

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

$$r_{t_i}^k = r_0 + \lambda(DL_{t_i}^k - 1) \tag{7}$$

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

$$\sum_{i=1}^{m} \varphi_i (r_0 + \lambda(N-1)) \leq \mathcal{B} \tag{8}$$

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.

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

V. DISTRIBUTED TASK SELECTION MECHANISMS

In this section, we first prove that the task selection problem is NP-hard, and then propose distributed task selection algorithms to help users select tasks while maximizing their total profits at each sensing round.

Theorem 1. *The task selection problem is 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.

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.

$$\begin{aligned} \max \quad & P(\mathcal{T}_{u_i}^k) = \sum_{t_j \in T_{u_i}^k} r_{t_j}^k \\ \text{s.t.} \quad & \Gamma_{\mathcal{T}_{u_i}^k} \leq B_{u_i}^k \end{aligned} \tag{10}$$

Given the graph G, the problem in Eq. 10 is to find a path originated at L_{u_i} with total travelling time no more than $B_{u_i}^k$ such that the total rewards gained from vertices is maximized. Hence we can see the problem in Eq. 10 is actually an orienteering problem [27] which is already proved to be NP-hard. Since problem in Eq. 10 is a special case of problem in Eq. 1 where $C(\mathcal{T}_{u_i}^k) = 0$, the task selection problem shown in Eq. 1 is also NP-hard. □

In the following, we propose an optimal dynamic programming based task selection algorithm and an efficient greedy task selection algorithm.

A. Dynamic Programming based Task Selection Algorithm

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 $dp[\mathtt{i}][j]$ denotes the shortest path for traveling the set of tasks in $\mathtt{i}$ starting from

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.

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

$$dp[\mathbb{i}][j] = \begin{cases} \min_{o(\mathbb{i})}\{dp[\mathbb{i}][j]_{o(\mathbb{i})}\} & t_j \in \mathbb{i}, \\ \infty & t_j \notin \mathbb{i}. \end{cases} \tag{11}$$

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

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. 11, we can have

$$dp[\mathbb{i} \mid 1 \ll (q-1)][q] = \min_{1\le j\le m}\{dp[\mathbb{i}][j] + dist[j][q]\} \tag{12}$$

From Eq. 12, 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. 12.
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.

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.

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. 12, 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 $dp[\mathbb{i}][j]$, it needs to run m steps according to Eq. 12. Therefore, the computational complexity of the dynamic programming based task selection algorithm is $O(m^2 2^m)$. □

B. Greedy Task Selection Algorithm

Although the dynamic programming based task selection algorithm can provide the optimal solution, it is not suitable for a large scale of tasks since its computational cost is too expensive. Therefore, we further propose an efficient greedy task selection algorithm.

We use the profit provided by the candidate tasks as a criteria, which is calculated as the reward of the task minus the cost of the movement from the current location to the location of the task. Thus, each mobile user will greedily select the task which can mostly increase the total profit at each step within the traveling time/distance budget until no satisfied task can be found.

Theorem 3. *The greedy task selection algorithm has a computational complexity of $O(m^2)$.*

Proof. When choosing the next task, users will consider all available tasks and the number of tasks at most equals m. Moreover, each user at most can select m tasks to perform. Hence the computational complexity of the greedy task selection algorithm is $O(m^2)$. □

VI. Performance Evaluation

In this section, we evaluate the performance of the proposed demand-based dynamic incentive mechanism and the task selection algorithms. We compare our algorithm with the steered crowdsensing mechanism [13] which dynamically changes the rewards of tasks according to expected quality improvements. As for the dynamic incentive mechanisms proposed in [11] and [15], 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.

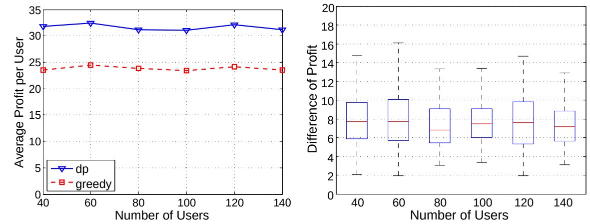
Steered crowdsensing mechanism: In [13], the reward of a task changes dynamically according to the expected quality improvements of the task. The reward function (Eq. 12 in [13]) of the steered crowdsensing mechanism is rewritten as follows.

$$R_{t_i}^k = R_c + \mu \Delta Q(x) \tag{13}$$

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. 13 looks similarly to our demand-based dynamic reward function in Eq. 7. 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-based 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-based 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.



(a) (b)

Fig. 5. The comparison of the dynamic programming based task selection algorithm and the greedy task selection algorithm.

In our experiments, the locations of mobile users and sensing tasks are randomly generated in a 3000m × 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.

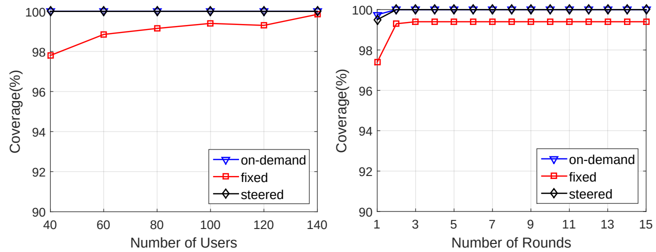
A. Comparison of The Task Selection Algorithms

We first compare the performance of the dynamic programming based task selection algorithm with the greedy task selection algorithm. Figure 5(a) show the average profit per user against the number of users at the sensing round 2. The average profit per user is the total profits of all users divided by the total number of users for 100 experiments. We can see that the optimal dynamic programming based task selection algorithm achieves higher profit than the greedy task selection algorithm. Figure 5(b) shows the boxplot of the difference of the user profit between the two task selection algorithms for all users in 100 experiments. We can see that the dynamic programming based task selection algorithm always obtains a higher profit for any user, and the difference varies from 2 to 16 in all experiments. Although the greedy task selection algorithm is not optimal, it is much faster than the dynamic programming based task selection algorithm, which can be used for crowdsensing with large scale. In the following experiments, we use the optimal dynamic programming based task selection algorithm.

B. Coverage

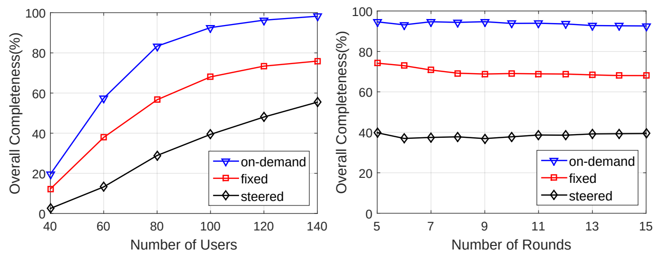
Coverage measures how good the algorithm balances the popularity among sensing tasks, which is a kind of spatial metric. The larger the coverage, the better the balance.

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-based



(a) (b)

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

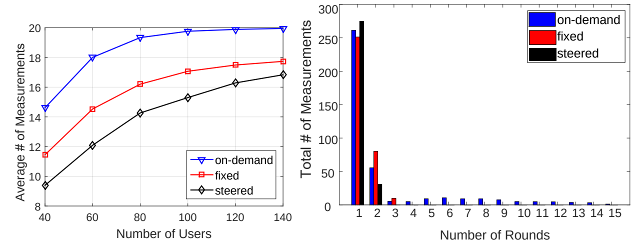


(a) (b)

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

incentive mechanism and the steered crowdsensing incentive mechanism always achieve better coverage than the fixed incentive mechanism. The demand-based 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-based 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-based 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.



(a) (b)

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

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.

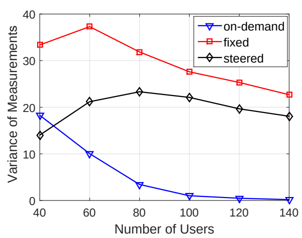
Impact of user number: Figure 7(a) shows the overall completeness of the three mechanisms against the number of users until the last sensing round. The overall completeness increases as the number of mobile users increases, as there are more users to work on the tasks. Compared to the fixed incentive mechanism and the steered incentive mechanism, the demand-based incentive mechanism has a higher overall completeness and the superiority becomes more obvious for more mobile users.

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-based incentive mechanism always has a higher overall completeness than the fixed incentive mechanism and the steered incentive mechanism for all sensing rounds. The demand-based incentive mechanism achieves almost 100% completeness while the fixed incentive mechanism only has about 70% completeness. The steered crowdsensing incentive mechanism has the worst performance that only achieves 40% completeness since it only considers the quality of tasks but does not take the deadline of tasks into consideration.

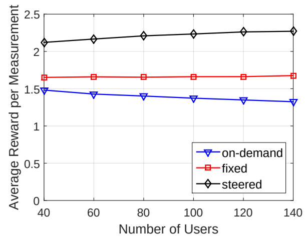
D. # of Measurements

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 100 mobile users. Even with only 60 users, the average # of measurements



(a)



(b)

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

can reach 18 for the on-demand incentive mechanism, while it is 14.3 and 12 for the fixed and steered incentive mechanism, respectively.

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. 13. 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 the 4th 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 much smaller than the other two incentive mechanisms. 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 more even distribution of measurements among tasks. However, when the number of users is small, the variance of measurements shows an increasing trend for the fixed and steered incentive mechanisms when the number of users increases. This is because the number of measurements of each task is too small (e.g., the average number of measurements when there are 40 users as shown in Figure 8(a), which results in a small variance of measurements. The on-demand incentive mechanism does not have this trend, which is because it takes the demands to tasks into consideration and encourage users to select far-away tasks to realize well balance of participation among tasks.

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.

Impact of user number: 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.

However, the decrease is slightly and the average reward per measurement when there are 40 users is not much higher than that when there are 140 mobile users. This is because we use the AHP framework to characterize the relative demand of one task over another instead of increasing or decreasing the rewards of tasks directly. Therefore, we can conclude that the proposed on-demand incentive mechanism is scalable to the number of users. As for the steered incentive mechanism, the average reward per measurement increases as the increasing of the mobile users. This is because that with the enough users, more tasks will be selected at the previous stages when the rewards of sensing tasks can be relatively higher.

VII. Conclusion

In this paper, we focused on location-dependent crowdsensing systems, and proposed a demand-based dynamic incentive mechanism that dynamically changes the reward of each task in an on demand way to balance the popularity among tasks. Extensive experiments show that the 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-based 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. of ACM MobiCom*, 2012, pp. 173–184.

[2] P. Dutta, P. M. Aoki, N. Kumar, A. Mainwaring, C. Myers, W. Willett, and A. Woodruff, "Common sense: participatory urban sensing using a network of handheld air quality monitors," in *Proc. of ACM SenSys*, 2009, pp. 349–350.

[3] "Sensorly," http://www.sensorly.com/.

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

[5] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: an end-to-end participatory urban noise mapping system," in *Proc. of ACM/IEEE IPSN*, 2010, pp. 105–116.

[6] I. Schweizer, C. Meurisch, J. Gedeon, R. Bärtl, and M. Mühlhäuser, "Noisemap: multi-tier incentive mechanisms for participative urban sensing," in *Proc. of PhoneSense*, 2012, pp. 1–5.

[7] T. Liu and Y. Zhu, "Social welfare maximization in participatory smartphone sensing," *Computer Networks*, vol. 73, pp. 195–209, 2014.

[8] J. Sun and H. Ma, "Heterogeneous-belief based incentive schemes for crowd sensing in mobile social networks," *Journal of Network and Computer Applications*, vol. 42, pp. 189–196, 2014.

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

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

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

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

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

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

[15] B. Guo, H. Chen, Z. Yu, W. Nan, X. Xie, D. Zhang, and X. Zhou, "Taskme: Toward a dynamic and quality-enhanced incentive mechanism for mobile crowd sensing," *International Journal of Human-Computer Studies*, 2016.

[16] Y. Chon, N. D. Lane, Y. Kim, F. Zhao, and H. Cha, "Understanding the coverage and scalability of place-centric crowdsensing," in *Proc. of ACM UbiComp*, 2013, pp. 3–12.

[17] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, "Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *Proc. of IEEE INFOCOM*, 2014, pp. 1231–1239.

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

[19] S. He, D.-H. Shin, J. Zhang, and J. Chen, "Toward optimal allocation of location dependent tasks in crowdsensing," in *Proc. of IEEE INFOCOM*, 2014, pp. 745–753.

[20] H. To, C. Shahabi, and L. Kazemi, "A server-assigned spatial crowdsourcing framework," *ACM Transactions on Spatial Algorithms and Systems*, vol. 1, no. 1, p. 2, 2015.

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

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

[23] T. Luo, H.-P. Tan, and L. Xia, "Profit-maximizing incentive for participatory sensing," in *Proc. of IEEE INFOCOM*, 2014, pp. 127–135.

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

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

[26] T. Saaty, "The analytic hierarchy process, Mcgraw-Hill, New York, 1980."

[27] A. Blum, S. Chawla, D. R. Karger, T. Lane, A. Meyerson, and M. Minkoff, "Approximation algorithms for orienteering and discounted-reward tsp," *SIAM Journal on Computing*, vol. 37, no. 2, pp. 653–670, 2007.