HyTasker: Hybrid Task Allocation in Mobile Crowd Sensing

Jiangtao Wang, Feng Wang, Yasha Wang, Leye Wang, Zhaopeng Qiu, Daqing Zhang, Bin Guo, Qin Lv

Abstract—Task allocation is a major challenge in Mobile Crowd Sensing (MCS). While previous task allocation approaches follow either the opportunistic or participatory mode, this paper proposes to integrate these two complementary modes in a two-phased hybrid framework called HyTasker. In the offline phase, a group of workers (called *opportunistic workers*) are selected, and they complete MCS tasks during their daily routines (i.e., opportunistic mode). In the online phase, we assign another set of workers (called *participatory workers*) and require them to move specifically to perform tasks that are not completed by the opportunistic workers (i.e., participatory mode). Instead of considering these two phases separately, HyTasker jointly optimizes them with a total incentive budget constraint. In particular, when selecting opportunistic workers in the offline phase of HyTasker, we propose a novel algorithm that simultaneously considers the predicted task assignment for the participatory workers, in which the density and mobility of participatory workers are taken into account. Experiments on two real-world mobility datasets demonstrate that HyTasker outperforms other methods with more completed tasks under the same budget constraint.

Index Terms—Mobile crowd sensing, task allocation, hybrid approach.

1 Introduction

With the proliferation of sensor-rich mobile devices, *Mobile Crowd Sensing* (MCS) [1], [2] has emerged as a new way of sensing, which has drawn much attention from both academia [3], [4] and industry [5], [6]. MCS has stimulated in a variety of environmental, commercial, and social applications [7], [8], [9], [10], [11], [13], [14], [50], where dynamically-moving citizens (called workers) contribute urban sensing information (e.g., traffic congestion status, air quality, and noise level) through mobile devices.

Task allocation or worker selection is one of the major challenges in MCS [12], which has a significant impact on the efficiency and quality of the sensing tasks [15], [18], [19], [20], [26]. Recently, there have been many studies on MCS task allocation, such as [21], [22], [23], [24], [25], [29], [30], [34], [36], [37], which can be divided into two categories based on the workers' movement patterns and participation mechanisms [2], [13], [15], [49]. (1) In the *opportunistic mode*, an MCS system assigns tasks to a number of selected workers, who will complete the tasks during their daily routines without the need to change their routes [21], [22], [23], [24], [25], [29], [30]. (2) In the *participatory mode*, however, workers are required to change their original routes and move specifically to certain places to complete MCS tasks [17], [34], [36], [37]. Existing MCS solutions adopt either the opportunistic mode or the participatory mode to tackle the task allocation problem. However, both modes have their own advantages and disadvantages, which we elaborate as follows.

The *opportunistic mode* does not require knowledge of the workers' intended travel routes, so it is less intrusive for the workers and less costly for the task organizers. However, the sensing quality of the assigned tasks depend heavily on the workers' routine trajectories. For tasks that are located at places visited by few or even no workers, their sensing quality can be very poor. Additionally, in order to select an optimal set of workers, task allocation strategies based on the opportunistic mode usually needs to predict the workers' trajectories, which significantly affects the optimality of the task allocation plan. Although different trajectory prediction algorithms [16], [25], [36] have been proposed and proved to be effective to some extent, their accuracy cannot be theoretically guaranteed due to complicated and unpredictable real-life conditions. Thus, the final sensing quality achieved for some tasks may be lower than expected.

The *participatory mode* requires workers to move specifically to task locations, which can guarantee task completion. However, since workers need to deviate from their original routines and travel to task locations, it incurs extra travel cost and can be intrusive to the workers. It also increases the task organizers' incentive cost, since the task organizers usually have to pay extra incentive rewards to compensate for the traveling cost of the workers. Moreover, task allocation strategies based on the participatory mode only utilize mobile users who are willing to change their routes and travel intentionally for the tasks. As reported by some recent studies on human factors in MCS [18], [20], a large proportion of mobile users are willing to contribute sensing data but do not want to change their routine trajectories. This group of mobile users is excluded from the candidate workers in the participatory mode, which is a waste of limited sensing resources.

- *Jiangtao Wang is with school of computing and communications at Lancaster University, UK.*
  E-mail: jiangtao19871104@sina.com;
- *Feng Wang, Leye Wang, Zhaopeng Qiu and Daqing Zhang are with School of EECS, Peking University, Beijing; Yasha Wang is with National Engineering Research Center of Software Engineering, Peking University. All of them are with Key Laboratory of High Confidence Software Technologies, Ministry of Education.*
- *Bin Guo is with Northwestern Polytechnical University.*
- *Qin Lv is with University of Colorado Boulder.*

Manuscript received June 30, 2017.

In this paper, motivated by the complementary nature of these two modes, we propose a hybrid MCS task allocation framework, called HyTasker, which effectively integrates the opportunistic mode and the participatory mode via a two-phased design. In the offline phase, we recruit a number of workers (called *opportunistic workers*) to complete sensing tasks during their routine trajectories. In the online phase, we further assign some other workers (called *participatory workers*) to locations where tasks cannot be completed by the opportunistic workers alone. Specifically, we study the typical budget-constrained MCS task allocation problem in this paper, where one task organizer launches a certain MCS campaign in the city with a number of location-dependent homogenous sensing tasks, with the goal of maximizing the number of completed tasks while keeping the total incentive rewards under a budget constraint.

Compared with the pure opportunistic or participatory mode, the advantage of HyTasker can be summarized in the following two aspects.

First, *from the perspective of the workers*, it naturally accommodates the workers' participation preferences and makes full use of the available human sensing resources. In real-world applications, workers' preferred way of participation can be different. For example, some office employees are busy all day and do not have time to take a detour for task completion. In this case, they only accept to complete tasks on their daily routine trajectories. In contrast, some retired or unemployed citizens who have plenty of leisure time may be willing to move intentionally and complete tasks to earn incentive rewards. HyTasker assigns tasks based on the workers' preferences, hence making better use of the potential human sensing resources in the city.

Second, *from the perspective of the task organizer*, HyTasker can achieve a better tradeoff between sensing quality and cost. Compared with pure participatory-mode approaches, it leverages some opportunistic workers to unintentionally complete tasks, which significantly reduces the incentive cost. In contrast to the pure opportunistic-mode approaches, it further improves the sensing quality by assigning some participatory workers to move and complete tasks in uncovered locations.

To illustrate how HyTasker works and further highlight the research challenges, we present a motivating example as follows. The city government launches an MCS campaign, called AirSense, for collecting real-time air quality information in different regions from 8:00am to 6:00pm every day in the city with a total budget constraint (e.g., 2000 USD per day). As the entire sensing area can be divided into 20 subareas (or called task locations), we can view the AirSense campaign as 20 homogenous sensing tasks in the same sensing period (e.g., from 8:00am to 6:00pm). 500 mobile users have registered as candidate workers in AirSense, and their historical records of connections to the cell towers are utilized by AirSense only for the purpose of task allocation after proper anonymization. According to their declared participation preferences on a specific day, 350 workers are candidate opportunistic workers, while 150 are candidate participatory workers. As the budget is limited, AirSense cannot recruit all candidates to complete all tasks, and its goal is to design an effective task allocation mechanism to maximize the number of completed tasks. Hence, AirSense adopts the HyTasker framework for task allocation, which has the following two phases. First, in the offline phase, HyTasker selects a set of opportunistic workers from 350 candidates, and each of them is given a fixed reward (e.g., 10 USD per worker) for the entire sensing period [22], [25]. The selected opportunistic workers will collect sensing data for all tasks during their routine trajectories when they connect to the cell towers. Then, in the online phase (e.g., one hour before the end of sensing period), HyTasker spends the rest of the budget to assign another set of participatory workers from 150 candidates with uncompleted tasks so far. The participatory workers will change their daily routes to move intentionally for task completion, and the incentive reward each worker gets is in proportion to his/her travel distance (e.g., 2 USD per kilometer) [34], [36].

Given the basic design of HyTasker in the motivating example above, we can see at least the following research challenges. First, the two types of workers (i.e., opportunistic workers and participatory workers) share a total incentive budget, thus we cannot consider them separately. A naïve solution is to try different proportions of the budget division, and then directly adopt the state-of-the-art task allocation methods for opportunistic mode and participatory mode, respectively. However, as we cannot determine an optimal budget division plan, such a naïve solution may not perform well. Therefore, more sophisticated methods are needed to jointly optimize the offline and online phases. Second, in the offline phase of opportunistic worker selection, we need to consider possible online task assignments for the participatory workers in the future, which is challenging since we cannot foresee the precise locations of the participatory workers and the completion status of tasks in the online phase.

In an effort to address the objectives and challenges mentioned above, our work makes the following contributions:

(1) By analyzing the complementary nature of the participatory-mode and opportunistic-mode MCS, we propose a two-phased hybrid task allocation framework, called HyTasker. It effectively integrates the two modes by selecting opportunistic workers in the offline phase and participatory workers in the online phase, via a joint optimization process. To the best of our knowledge, we are the first to combine these two modes in the MCS task allocation problem.

(2) We propose a nested-loop greedy process to select the opportunistic workers by pre-considering future online task assignment of the participatory workers, which consists of two key mechanisms that are not adopted in the state-of-the-art MCS solutions. First, by considering the historical density of the participatory workers, HyTasker assigns higher priority to the opportunistic workers who are not only capable of completing more tasks but also can complete uncovered tasks in areas with fewer participatory workers. Second, it records each local-optimal subsets obtained during the greedy process, and further selects the optimal one by simultaneously predicting the task assignments of the participatory workers.

(3) We evaluate HyTasker extensively using D4D [40] and Gowalla [52], two real-world open datasets with a large number of users' mobility traces. The experimental results demonstrate that HyTasker outperforms other methods with more completed tasks under the same budget constraint.

2 RELATED WORKS

A number of research works exist for selecting MCS workers, who can complete MCS tasks during their daily routines without the need to change their original trajectories. One group of studies considered the worker selection of a single MCS task with certain goals and constraints [21], [22], [23], [24], [25], [27], [29]. For example, the authors studied worker recruitment for a single MCS task, and they proposed different recruitment strategies to select a predefined number of workers so as to maximize the task's sensing quality [21], [22], [23], [24], or select a minimum number of workers to ensure a certain level of sensing quality [25], [29]. Another group of studies attempted to optimize the overall utility of multiple concurrent sensing tasks in a multi-task-oriented MCS platform, where tasks share the limited resources [30], [31], [32], [53]. For example, both [30] and [31] proposed multi-task allocation algorithms to maximize overall system utility when the tasks share a limited incentive budget. The multi-task allocation strategy proposed in [32] aims to optimize the overall utility when multiple tasks share a pool of workers with a sensing bandwidth constraint.

Another category of MCS tasks require workers to change their original routes and specifically move to certain places. There are two models for task publishing, i.e., worker selected tasks (WST) [33] and server assigned tasks (SAT) [34], [36], [35], [37], [38], in which tasks are selected by workers themselves or automatically assigned by the server, respectively. Our hybrid task allocation problem follows the SAT model. Prior studies in the SAT model [34], [36], [37], [38] assigned existing workers to tasks in the MCS system with various optimizing goals and constraints. For instance, the authors of [34], [36] aimed to maximize the number of completed tasks or overall task quality on the server side, while ensuring constraints on workers' maximum number of accepted tasks and task completion regions. The objective of [38] is to minimize the traveling cost for completing a set of given tasks while seeking solutions that are socially fair.

The above studies adopt either opportunistic or participatory mode for MCS task allocation, while our work proposes a hybrid solution to achieve a better tradeoff between sensing quality and cost. Technically, our defined problem is more challenging, as we have to jointly optimize these two modes of task allocation with a shared incentive budget constraint. Different from existing MCS worker selection approaches, we develop a novel opportunistic worker selection algorithm with two unique mechanisms. First, instead of selecting workers who can only complete more tasks or cover larger areas [22], [25], [31], [30], HyTasker assigns higher priority to those who can complete more tasks in participatory-worker-sparse areas. Second, while existing worker selection approaches commonly end when the total budget has been used up and output the final set of selected workers [22], [25], [31], [30], HyTasker further records, estimates, and selects the best set of workers by simultaneously considering the predicted task assignments of the participatory workers.

In summary, although classifying workers into opportunistic and participatory group is common in MCS, HyTasker is novel in the following aspects: (1) Difference in basic idea. The basic idea of HyTasker is to utilize these two groups to achieve a good trade-off between sensing quality and cost, as they are complementary. (2) Difference in overall framework. State-of-the-art research works proposed task allocation methods for two types of workers separately. In contrast, HyTasker designs a two-phased approach by leveraging two types of supporters, where the online and offline phases are jointly considered as a total budget is shared. (3) Difference in core algorithm. Different existing work, the key technical challenge of HyTasker lies in the offline opportunistic worker selection, which requires the joint consideration of future possible task assignment of the participatory workers. To address this challenge, we propose a nested-loop process, which is the key technical contribution of HyTasker.

3 HYTASKER: SYSTEM OVERVIEW

In this section, we first analyze and formulate the hybrid task allocation problem, then describe the proposed HyTasker framework to solve this problem.

3.1 Problem Analysis and Formulation

Similar to some previous studies such as [15], [25], [27], this paper focuses on the task allocation of homogenous MCS tasks for urban environmental sensing. Specifically, the application scenario is that one organizer launches a certain type of MCS campaign (e.g., air quality sensing) during a certain period of time under a total budget constraint B. The entire sensing area can be divided into n subareas, and the data collection mission in each subarea is defined as a "task" in this paper. Thus, the MCS campaign consists of n location-dependent homogenous sensing tasks $T = \{t_1, t_2, \ldots, t_i, \ldots, t_n\}$ during the same sensing period (e.g., 8:00am-6:pm). A task can be completed once a recruited worker moves into the corresponding subarea during the sensing period. The goal of HyTasker is to maximize the number of completed tasks by recruiting both opportunistic and participatory workers. In fact, sensing quality of MCS tasks is actually influenced by many factors (e.g., the quality of workers and number of reports for each task). Since the focus of our work does not lie in the design of task quality model, we follow the assumption of many existing works [17], [22], [25], [29], [34] to simplify the problem formulation, that is, we assume that the worker's quality is equal and one report means that a task is completed. Thus, our objective is to maximize the number of completed task with a predefined budget constraint.

We divide the candidate workers into two disjoint categories $OW = \{ow_1, ow_2, ...ow_j...ow_l\}$ and $PW = \{pw_1, pw_2, ...pw_k...pw_m\}$ based on their self-defined preferences of participation mode. 1)*The candidate opportunistic workers,* denoted as OW, would complete sensing tasks during their routine trajectories. 2)*The candidate participatory* *workers.* This group of workers, denoted as PW, are more active and are willing to change their routes to complete sensing tasks. They actively report the participation information to the cloud server, including their current online location $P = \{p_1, p_2, ...p_k...p_m\}$, spatial region $RW = \{rw_1, rw_2, ...rw_k...rw_m\}$ within which they are willing to travel. Each of them can get incentive rewards in proportion to the actual travel distance online, which is denoted as $I * Distance(pw_k, PT_k)$, where I is the proportion and $Distance(pw_k, PT_k)$ is the length of the shortest path for pw_k to complete assigned task set PT_k. The maximum number of tasks pw_k can be assigned is denoted as L.

Based on the problem analysis above, we formally define the hybrid task allocation problem as follows. 1)*In the offline phase*, our objective is to select a subset of candidate opportunistic workers $OW_f \subseteq OW$ and pay each of them a fixed and equal incentive reward I_c. Similar to [25], [15], [27], [32], HyTasker adopts a piggyback MCS paradigm for the opportunistic workers, in which the workers will complete sensing tasks online when they connect to corresponding cell towers or perform check-in in certain subareas. 2)*In the online phase*, at timestamp TS, we aim to assign the participatory workers some tasks that are not completed by the opportunistic workers so far. We denote the full set of task-and-worker pairs as $V = PW \times T = \{(pw_k, t_i)|pw_k \in PW, t_i \in T\}$. So the objective of online task assignment is to select a subset V_f of V subject to the constraints (i.e., spatial region and maximum number of assigned tasks). Here, similar to previous studies such as [45], [46], [47], we assume that once a participatory worker pw_k is assigned a task, the probability (called task acceptance rate) that s/he will accept the task is ac_k, which has already been learned from his/her previous participation history.

It is important to note that the online phase and the offline phase are correlated, because they share a total budget constraint. Specifically, for the MCS platform, the optimization goal is to maximize the total number of completed task set, denoted as $T_c(T_c \subseteq T)$, while keeping the total incentive reward under the budget constraint. The optimization problem can be formulated as follows:

$$Maximize\ |T_c| \tag{1}$$

$$Subject\ to: I_c * |OW_f| + \sum_{pw_k \in PW} I * Distance(pw_k, PT_k) \leq B \tag{2}$$

3.2 HyTasker Overview Design

As illustrated in Fig. 1. the design of HyTasker mainly includes two phases: *In the offline phase*, it selects a set of *opportunistic workers*, and each of them is paid with a fixed reward for the entire sensing period. These workers will complete sensing tasks online during their daily routine. *In the online phase*, HyTasker further spends the rest of the budget by assigning another set of *participatory workers* with certain tasks that have not been completed by the opportunistic workers. The participatory workers will move intentionally to the sensing locations to finish the tasks and get incentive rewards in proportion to their travel distances. The key points of these two phases are summarized as follows, respectively.

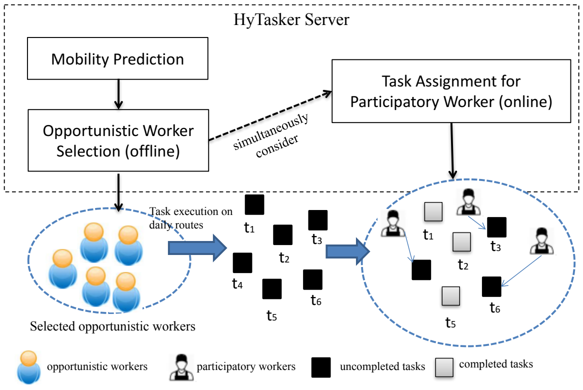

HyTasker Server
Mobility Prediction
Opportunistic Worker Selection (offline)
Task Assignment for Participatory Worker (online)
simultaneously consider
Task execution on daily routes
Selected opportunistic workers
t1
t2
t3
t4
t5
t6
opportunistic workers
participatory workers
uncompleted tasks
completed tasks

Fig. 1. The overview design of HyTasker. In the offline phase, a group of workers (called opportunistic workers) are selected, and they complete MCS tasks during their daily routines; In the online phase, we assign another set of workers (called participatory workers) and require them to move specifically to perform tasks that are not completed by the opportunistic workers.

- *Opportunistic workers selection (offline).* To solve the above-defined hybrid task allocation problem, the biggest challenge lies in the offline selection of the opportunistic workers. Since the opportunistic workers and the participatory workers share a total budget constraint, we cannot consider them separately. Instead, when selecting the opportunistic workers offline, we must consider the future online task assignments for the participatory workers. But the challenge is that, during the offline phase of opportunistic worker selection, we cannot foresee the precise locations of the participatory workers and the completion status of tasks in the online phase. To address this, we propose a heuristic greedy based opportunistic worker selection algorithm by simultaneously considering the online task assignments of the participatory workers. The main process and basic idea of this phase will be presented in Section 3.3, and the detailed algorithms will be illustrated in Section 4.
- *Task assignment for the participatory workers (online).* Note that when selecting the opportunistic workers offline, the task assignments of the participatory workers are predicted but are not actually delivered to the workers. Thus, in the online phase, we will re-do the task assignments based on real-time location information reported by the candidate participatory workers and the actual task completion status. The goal is to cover locations where the tasks are not completed by the opportunistic workers. Since the online task assignment problem for the participatory workers is well studied in [34], [15], we directly adopt the maximum-flow based algorithms in [34], [15] and focus the remaining text on the offline phase above.

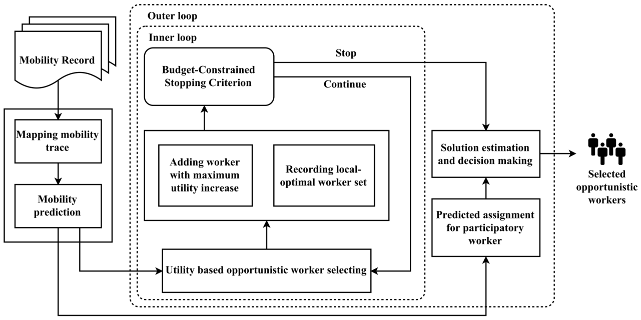

Outer loop
Inner loop
Mobility Record
Mapping mobility trace
Mobility prediction
Budget-Constrained Stopping Criterion
Stop
Continue
Adding worker with maximum utility increase
Recording local-optimal worker set
Utility based opportunistic worker selecting
Solution estimation and decision making
Predicted assignment for participatory worker
Selected opportunistic workers

Fig. 2. Opportunistic worker selection in the offline phase: Main process)

3.3 Opportunistic Worker Selection: Main Process and Basic Idea

The key technical challenge of HyTasker lies in offline opportunistic worker selection, which requires the joint consideration of future possible task assignment of the participatory workers. To address this challenge, we propose a nested-loop process (as shown in Fig. 2).In this process, the inner-loop searches for and determines which opportunistic worker should be selected, while the outer-loop process determines the best set of workers by estimating the number of completed tasks for a given subset of opportunistic workers and predicting the task assignments of the participatory workers. Finally, the output is the subset of opportunistic workers that achieves the maximum estimated number of completed tasks.

The proposed process above is more technically challenging and complicated than the state-of-the-art studies for opportunistic worker selection (such as [22], [25], [31], [30]) in the following two aspects.

First, for the inner loop, the challenge is how to select the most beneficial opportunistic workers. Existing studies incrementally select workers with the highest estimated coverage gain [22], [25], [31], [30]. However, HyTasker should further consider the future online task assignments of the participatory workers.

Specifically, this inner-loop process is designed based on a key idea, that is, we prefer to select opportunistic workers with two characteristics: (1) workers who can complete more tasks; (2) workers who can complete tasks located in areas where the participatory workers are sparsely distributed according to their historical mobility records. As illustrated by the example in Fig. 3, we assign higher priority to the opportunistic worker B than A, because B is more likely to complete tasks within areas with fewer participatory workers. In this paper, we introduce the concept of location entropy [41] in social network community to realize this idea, which will be described in Section 4.2 with more details.

Second, for the existing work [22], [25], [31], [30], the worker selection process ends when the total budget has been used up, and the finally obtained set of workers is the output. However, in our problem, it is not optimal if we spend all the budget on the opportunistic workers. Therefore, after adding and selecting one worker, we record the obtained subset of opportunistic workers as a snapshot. Ultimately, we determine which subset (snapshot) should be selected by simultaneously considering the predicted task assignments of the participatory workers online. This component will be illustrated with more details in section 4.3.

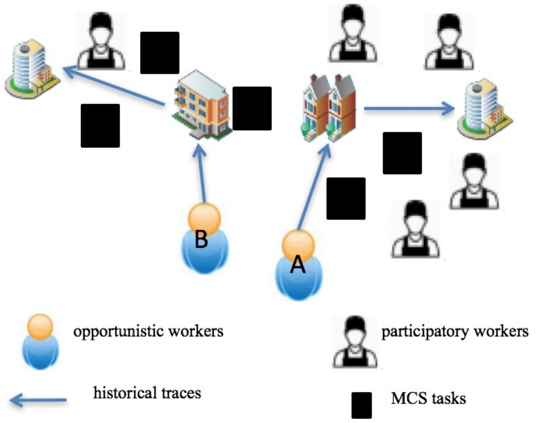


Fig. 3. An example to illustrate the priority of opportunistic workers (We assign higher priority to worker B than A, because B is more likely to complete tasks in areas where the participatory workers are sparsely distributed)

4 CORE SUPPORTING ALGORITHMS

To implement the opportunistic worker selection process in Fig. 2, we need to further design three supporting algorithms. First, we need to profile and predict the mobility of all candidate workers (Section 4.1). Second, for the inner loop, we should determine which worker should be selected (Section 4.2). Third, as the execution of each iteration forms a feasible solution (one snapshot), we should further determine which one is optimal (section 4.3). Finally, we present the algorithm complexity analysis in Section 4.4.

4.1 Mobility Profiling and Prediction

Similar to [25], [31], [32], this paper assumes that the opportunistic workers will complete tasks when they connect to the cell tower or perform check-in. Thus, this algorithm predicts the probability of each worker connecting to different towers at least once during the sensing period. We count the average number of connections by each worker w_u at each cell tower (or check-in by each worker at each subarea) l_i, which is denoted as $\lambda_{u,i}$. For example, we set the entire sensing period as one day. To estimate $\lambda_{u,i}$ for of a specific day, we count the average number of connections or check-ins by w_u at l_i during each day in the historical mobility and connection records. Assuming that the connection or check-in sequence follows an inhomogeneous Poisson process [25], [31], [32], the probability of worker w_u connecting to cell tower (or performing check-in at the subarea) l_i for h times during a specific day can be modeled as:

$$\varphi_{u,i}(h) = \lambda_{u,i}^{h} * e^{-\lambda_{u,i}}/h! \tag{3}$$

Therefore, we can estimate the probability of worker w_u connecting or checking in at least once during a day at l_i as follows:

$$Pro_{u,i} = \sum_{h=1}^{\infty} \varphi_{u,i}(h) = 1 - e^{-\lambda_{u,i}} \tag{4}$$

Thus we predict the probability of a candidate opportunistic worker w_u completing task at l_i as:

$$\alpha_i(w_u) = 1 - e^{-\lambda_{u,i}} \tag{5}$$

4.2 Utility-based Candidate Solution Generation

HyTasker iteratively selects the most beneficial opportunistic worker, and the pseudocode of this process is presented in Algorithm 1. First of all, HyTasker prefers to select opportunistic workers who will visit as many task locations (subareas) as possible, which is similar to traditional worker selection studies. Besides this criteria, with the hybrid task allocation paradigm that we propose, HyTasker also prefers to select opportunistic workers who will visit locations that are far from the participatory workers, because this reduces the traveling cost of online task allocation for the participatory workers. For example, if two opportunistic workers are predicted to visit the same number of locations, we will prefer the one who can visit locations where the participatory workers are sparsely distributed.

Thus, HyTasker assigns priority to each task location by considering the past visits of participatory workers. Intuitively, the higher the number of visits to a task location, the lower the priority we assign to that task location. One naïve method is to set the priority of each task location as inversely proportional to the total number of historical visits of all participatory workers. However, this naïve measurement neglects the distribution of visits among different participatory workers, which should also be considered. The reason is that, if the visits belong to a small set of users, it is a bit risky for the online task allocation to count on this small proportion of frequently-visiting users. During the online phase, these users may be too far away or even decline to accept tasks. In summary, we should assign higher priority to task locations with fewer total visits and more concentrated (less uniform) distribution of visits. Based on the above observation, we introduce the concept of *location entropy* to characterize the priority of each task location, which considers both the total number and the distribution of the visits among participatory workers. A location would have higher entropy (lower priority) if there are many visits and the visits are distributed more evenly among participatory workers. In contrast, a location will have lower entropy (higher priority) if there are fewer total visits or the distribution of the visits is restricted to only a few participatory workers.

The location entropy of a task t_i is defined as follows:

$$Entropy(t_i) = -\sum_{pw \in PW_i} \frac{|Count_{pw,t_i}|}{|Count_{t_i}|} \times log \frac{|Count_{pw,t_i}|}{|Count_{t_i}|} \tag{6}$$

where PW_i denotes the set of participatory workers visited the tower which t_i belongs to, $Count_{t_i}$ denotes the times of the tower which t_i belongs to was visited by PW_i, and $Count_{pw,t_i}$ denotes the times of worker pw visited the tower which t_i belonged to.

Inspired by this concept, we formally define the priority of each task based on its location entropy as follows:

$$Weight(t_i) = \frac{1/Entropy(t_i)}{\sum_{t \in T}(1/Entropy(t))} \tag{7}$$

Then, the utility increase of adding one candidate opportunistic worker is calculated as follows:

$$\begin{aligned} &Utility(OW_f \cup \{ow_j\}) - Utility(OW_f) = \\ &\sum_{t_i \in T} Weight(t_i) \times \varPhi(i, OW_f \cup \{ow_j\}) \\ &- \sum_{t_i \in T} Weight(t_i) \times \varPhi(i, OW_f) \end{aligned} \tag{8}$$

where $\varPhi(j, OW_f)$ will be illustrated with more details in Eq.(9).

Algorithm 1 Utility-based candidate solution generation

```
Input: candidate opportunistic workers $OW$; candidate participatory workers $PW$; total budget constraint $B$.
Output: the set of candidate opportunistic worker set $SS$.
1: set $OW_f^{'} = \varnothing$
2: while $|OW_f^{'}| * I_c < B$ do
3:     set $MaxUtility = 0$
4:     for each $ow_j \in OW/OW_f^{'}$ do
5:         if $Utility(OW_f^{'} \cup \{ow_j\}) > MaxUtility$ then
6:             $MaxUtility = Utility(OW_f^{'} \cup \{ow_j\})$    ▷ $Utility$ is be defined in Eq.(8)
7:             $BestW = ow_j$
8:         end if
9:     end for
10:    $OW_f^{'} = OW_f^{'} \cup \{ow_j\}$
11:    $OW = OW - \{ow_j\}$
12:    append $OW_f^{'}$ to $SS$
13: end while
       return $SS$
```

4.3 Solution Estimation and Decision Making

At the end of each iteration, we record a snapshot, i.e., a subset of candidate opportunistic workers(see line 12 in Algorithm 1). Since we assume that each selected opportunistic worker will be given the same reward, the total number of iterations will be $\lfloor \frac{B}{I_c} \rfloor$. In other words, we get $\lfloor \frac{B}{I_c} \rfloor$ possible solutions for the opportunistic worker selection. We can then compare the optimality of those $\lfloor \frac{B}{I_c} \rfloor$ solutions based on the following steps:

First, we predict the probability of each task to be completed by a given subset of opportunistic workers. In Eq.(5), we have already obtained the probability of an opportunistic worker completing a given task. Thus, the probability that a task t_i can be completed by a set of selected opportunistic workers OW_f is defined as:

$$\varPhi(i, OW_f) = 1 - \prod_{ow_i \in OW_f} (1 - \alpha_i(ow_i)) \tag{9}$$

Second, we predict the optimal task assignments of the participatory workers by leveraging the maximum-flow-based algorithms [15], [34]. In [15], [34], the assumption

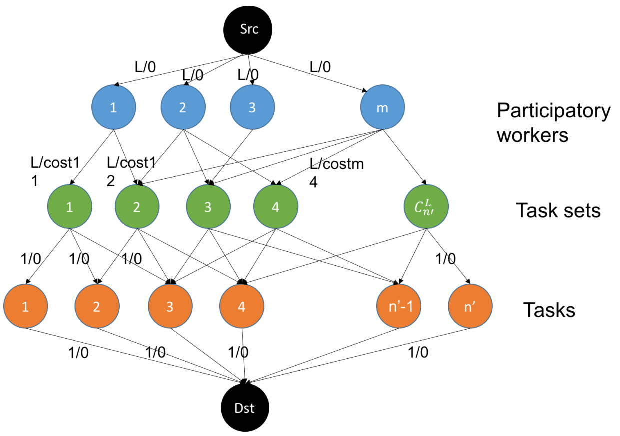


Fig. 4. The structure of maximum-flow network in a certain round of Monte-Carlo simulations (blue nodes: participatory workers; green nodes: task sets; orange nodes: tasks.)

is that the locations of workers are known and no task is completed . In contrast, in the offline worker selection phase of HyTasker, we only know the prediction probability of workers' locations and the task completion status. Therefore, we cannot directly adopt existing participatory worker assignment algorithm to address this issue. Hence, we adopt multiple rounds of Monte-Carlo simulations to tackle it. Specifically, in each round of simulation, we first generate the completion status of each task and the location of each participatory worker based on the probability. Then, we construct the network of the maximum-flow based algorithm based on the generated worker location and task completion status, and sequentially adopt the maximum-flow-based algorithm in [15] to perform task assignment. The average number of completed tasks for multiple rounds of simulations is regarded as the final results. Fig. 4 shows the network structure of the maximum-flow based algorithm. This network contains three levels of nodes: participatory workers, task sets, and tasks, and the edges linking different nodes are associated with specific capacity and cost of the flow. On the edges linking the blue and green nodes, the cost is the total distance of the shortest path for a specific participatory worker to complete a certain task set. For details about how the shortest path is obtained and how the maximum flow algorithm is executed, interested readers can refer to [15]. The pseudocode of the above predicted task assignment of participatory workers is presented in Algorithm 2.

Third, we estimate the total number of completed tasks by both the opportunistic workers and the participatory workers. The subset of opportunistic workers with maximum estimated total number of completed tasks will be selected as the final output of the offline phase.

4.4 Algorithm Complexity Analysis

In this section, we analyze the time complexity of the proposed opportunistic worker selection algorithm. The inner-loop process needs to estimate the utility increase of all unselected opportunistic workers can make. The running time complexity of the inner-loop process of each iteration will be $O(|T| \times |OW|)$. After each iteration selects the best

Algorithm 2 Task Assignment of Participatory Workers (offline prediction)

```
Input: candidate participatory workers $PW$; budget constraint $B_p$; tasks $T$; selected opportunistic workers $OW$
Output: the selected subset $V_f$; number of complete tasks $TW$
calculate probability of $t_i \in T$ completed by $OW$
$CompletedTasks = 0$
for i=0;i¡$Rounds$;i++ do
    simulate to generate the completed task $CT \subseteq T$
    simulate to generate participatory workers' location
    simulate if $pw_k \in PW$ is wiling to accept tasks based on the acceptance rate $ac_k$.
    select $C^L_{|CT|}$ task-task sets from task set $T - CT$
    calculate the shortest paths and travel cost
    construct the flow network $G = (V, E, C, W)$
    initialize flow $f$ to 0
    while t dohere exists an augmenting path in the residual network $G_f$
        select the augmenting path $p^*$ with minimum cost
        $c_f(p^*) = L$
        augment flow $f$ along $p^*$ with $c_f(p^*)$
    end while
    $CompletedTasks = CompletedTasks + f + |CT|$
end for
    return $CompletedTasks/Rounds$
```

opportunistic worker, the task assignment of the participatory workers will consume $O(|PW| \times L \times (|PW| + |T| + C^L_{|T|}))$[15]. The outer-loop process will be run $\lfloor \frac{B}{I_c} \rfloor$ times, so the time complexity is $O(\lfloor \frac{B}{I_c} \rfloor \times (|T| \times |OW| + |PW| \times L \times (|PW| + |T| + C^L_{|T|})))$.

5 EXPERIMENTAL EVALUATION

5.1 Experimental Purposes and Baselines

The goal of our experiments is to compare the performance of HyTasker and other baseline methods under different situations, such as different number of tasks, different number of workers, different total incentive budget, and so on. The performance comparison metric is the number of completed tasks.

Specifically, we provide the following baseline task allocation methods for comparative studies.

OPP-Greedy and OPP-GA *(two opportunistic-mode-based approaches)*: They only use opportunistic mode to maximize the number of completed tasks while keeping the budget constraint. They spend all budget to select the opportunistic workers. Here, we implement two representative algorithms as baselines, that is, OPP-Greedy and OPP-GA. Similar to [22], [25], [27], OPP-Greedy iteratively selects workers offline with the maximum utility increase until the total budget is used up, and the selected workers will complete tasks online during their daily routines. Similar to [26], OPP-GA use genetic algorithm (GA) to select the best set of opportunistic workers. These two baseline methods are used to test whether the hybrid approach is more effective than the pure opportunistic-mode approach.

PAR-Greedy and PAR-MaxFlow *(two participatory-mode-based approaches)*: They only use participatory mode to maximize the number of completed tasks while keeping the budget constraint. They spend all budgets to allocate the participatory workers online, and the workers will intentionally move to the task locations. Here, we select two representative algorithms as baselines, that is, PAR-Greedy and PAR-MaxFlow. Specifically, similar to [33], [37], PAR-Greedy iteratively selects worker-task pairs with the maximum utility increase until the total budget is used up. PAR-MaxFlow adopts the maximum-flow based algorithm in [34] to select the optimal set of task-worker pairs. These two baselines are designed to evaluate whether the hybrid approach is better than the pure participatory-mode approach.

BP-Hybrid *(budget-partition-based hybrid approach)*: This algorithm tries to divide the budget into different proportions, and then directly adopts the state-of-the-art task allocation methods for opportunistic mode and participatory mode, respectively. For each round, it first randomly generates the proportion of the budget division. Then, it uses the greedy algorithm in [25] to select the opportunistic workers and adopts the maximum-flow based algorithm in [34] to perform online task assignment for the participatory workers. BP-Hybrid repeats the above process for $\lfloor \frac{B}{I_c} \rfloor$ rounds, and the maximum number of completed tasks achieved is chosen as the final result.

5.2 Datasets and Experimental Setups

Two open mobility datasets, i.e., D4D [40] and Gowalla [52], are used to in the evaluations.

D4D Dataset. It contains two types of data records in Ivory Coast. One contains the information about cell towers, including tower id, latitude and longitude. The other one contains 50,000 users' phone call records. We select users randomly every 2 weeks (for weekdays) with anonymized ids and in total 10 sets of ten-day period of records are stored in the dataset. Here, for each set of ten-day records, we use the first nine-day records to model users' mobility patterns (described in Section 4.1), and use the 10th day as the test sensing period to execute the task assignment algorithm and evaluate the number of completed tasks. Specifically, we extracted records of the downtown area (100 cell towers with 1000 mobile users), as shown in Fig. 5.

Gowalla Dataset. It contains the check-in records of large number of users, including user-id, check-in-time, latitude, longitude, and location-id. To make sure the users' movements are correctly detected, we employ part of the original Gowalla dataset. In the employed dataset, the users are distributed in 20km × 30km rectangle regions except those in the sea (see Fig. 5.). Besides, we assume that the entire sensing area is divided into 150 equal-length virtual subareas (i.e., 2km2km per subarea). By removing the subareas in the sea, there are 128 subareas considered in the experiment. As the mobility trajectory in the Gowalla dataset is sparser than that of D4D dataset, we need longer time span to model the mobility pattern. Here, we divide the entire dataset into multiple subsets, and each subset contains one-month check-in records. For each subset, we use the last day of the month to test task assignment algorithm and evaluate the number of completed tasks, while using records of other prior days in this month to model workers' mobility pattern. The average performance for all subsets is the experimental results.

We need to set a number of parameters in the experiments, which are divided into task-relevant and worker-relevant as follows.

For the tasks, their locations are randomly distributed to a group of subareas within the target area, and there may be several tasks located in the same subarea. In the experiments, the sensing range of each task is within its deployed subarea's range. Similar to [25], [31], [32], the selected opportunistic workers can complete tasks in piggyback manner [28] when they connect to the cell towers (for D4D dataset) or perform a check-in (for Gowalla dataset). Moreover, each task is assumed to last for one day from 8:00am to 6:00pm and can be completed at any time within this period. Here, to simplify the problem, we assume that the online task assignment for the participatory workers is executed one hour before the end of the sensing period, and the participatory workers can complete all assigned tasks before the end of the sensing period.

For the workers, the settings are different for the opportunistic workers and the participatory workers. Each mobile user is set as either a candidate opportunistic worker or a candidate participatory worker with a certain probability (i.e., opportunistic worker: γ, participatory worker: $1 - \gamma$).The opportunistic workers are selected from candidates who make phone calls near these cell towers (for D4D) or perform check-ins within certain subareas (for Gowalla). For the participatory workers, the maximum number of tasks they can perform is randomly set to be between 2 to 5. Their initial locations, when the online task assignment is performed, are set as the most frequently-visited places in their historical mobility records. Moreover, the spatial region of each participatory worker is set as a rectangular region bounded by his/her historically-visited areas. Similar to [46], [47], we also simulate the acceptance rate of each participatory worker with a Gaussian distribution (with mean value μ), and further test the performance of different approaches by varying μ.

The reward for each opportunistic worker is set to 10 US dollars for the entire sensing period (i.e., one day), while the reward per kilometer for the participatory workers is set to 10 US dollars. Here we use the Manhattan Distance [42] to measure the travel distance between two locations.

The aforementioned parameter settings are summarized in Table 1. We carried out the experiments using a laptop computer with an Intel Core i7-4710HQ Quad-Core CPU and 16GB memory. HyTasker and other baseline methods were implemented with the Java SE platform on a Java HotSpotTM 64-Bit Server.

5.3 Experimental Results

In this section, we first report the experimental results of different methods and compare their performance with regard to the number of completed tasks.

5.3.1 Different value of total budget

In Fig. 6, we compare the performance of different methods under various settings of the total budget. In order to control

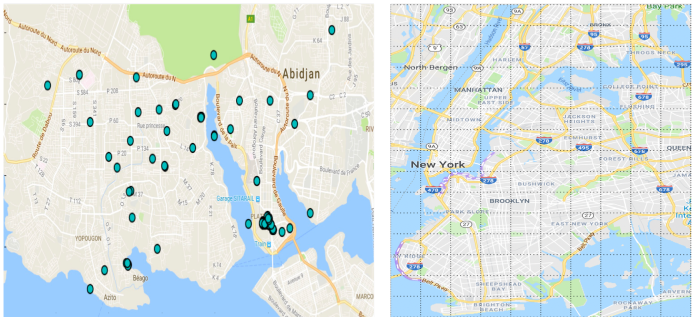


Fig. 5. Entire sensing area and the distribution of cell towers or subareas in our experiment (left for D4D, right for Gowalla)

TABLE 1
Summary of Experimental Parameter Settings

Parameters	Settings
Total budget	200,400,600,800,1000
Number of workers	100,200,300,400,500
Number of tasks	30,60,90,120,150
Percentage of OW (γ)	0.1,0.2,...,0.9
Mean value of acceptance rate μ	0.2,0.4,0.6,0.8,1.0
Maximum number of tasks for PW	Random generate between 2 and 5
reward per kilometer for PW	10
reward for each opportunistic worker	10

other variables, we fix the number of tasks at 90, the number of workers at 300, and the value of γ at 0.6. Here, we assume that the participatory workers will always accept the assigned tasks. From Fig. 6, we can see that the number of completed tasks increases with the total incentive budget for all methods, because a higher budget allows more workers to be recruited to complete more tasks. For the number of completed tasks, HyTasker outperforms other baseline methods in all budget settings and on both two datasets.

5.3.2 Different number of tasks

In Fig. 7, we compare the performance of different methods under different number of tasks. Here we fix the total incentive budget at 800 US dollars, the number of workers at 300, and the value of γ at 0.6. Here, we assume that the participatory workers will always accept the assigned tasks. From Fig. 7, we can see that HyTasker outperforms other baseline methods in all settings of the number of tasks and on both two datasets.

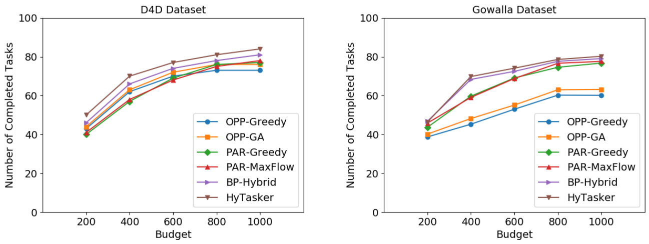


Fig. 6. Performance comparison under different settings of total budget

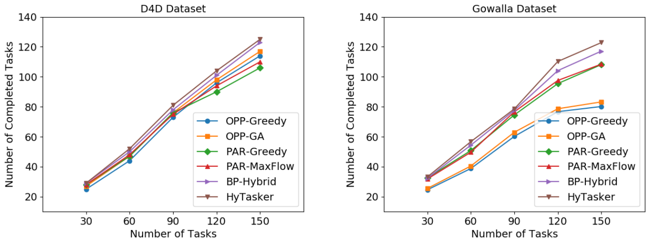


Fig. 7. Performance comparison under different number of tasks

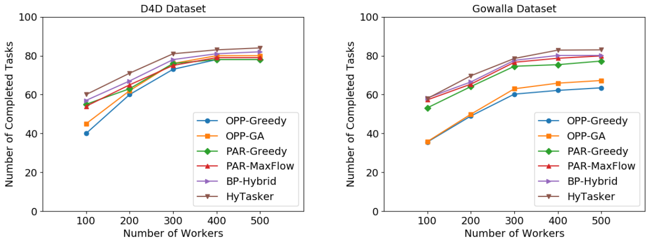


Fig. 8. Performance comparison under various number of workers

5.3.3 Different number of workers

In Fig. 8, we present the performance comparison under different number of workers. Here we fix the total incentive budget at 800 US dollars, the number of tasks at 90, and the value of γ at 0.6. From Fig. 8, we can see that the number of completed tasks increase when there are more candidate workers for all methods, because the task allocation algorithms have more candidate workers to choose from and can generate a better allocation plan. We can also see that HyTasker outperforms other baseline methods in all settings of the number of workers and on both two datasets.

5.3.4 Different value of γ

In Fig. 9, we illustrate the performance comparison under different values of γ. Here we fix the total incentive budget at 800 US dollars, the number of tasks at 90, and the number of workers at 300. We assume that the participatory workers will always accept the assigned tasks. From Fig. 9, we can also see that HyTasker outperforms other baseline methods in all settings and on both two datasets. We can also see that with the increase of γ, OPP-Greedy and OPP-GA can achieve a better performance for the number of completed tasks, while the performance of PAR-Greedy and PAR-MaxFlow become worse. This is because, with the fixed total number of candidate workers, there are more candidate opportunistic workers to choose from, while the number of candidate participatory workers becomes fewer.

5.3.5 Different value of μ

Similar to [46], [47], we assume that the acceptance rate of each participatory worker follows a Gaussian distribution with a mean value of μ, and present the performance of different approaches in Fig 10 by varying μ. Here we fix the total incentive budget at 800 US dollars, the number of tasks at 90, the number of workers at 300, and the value of γ at

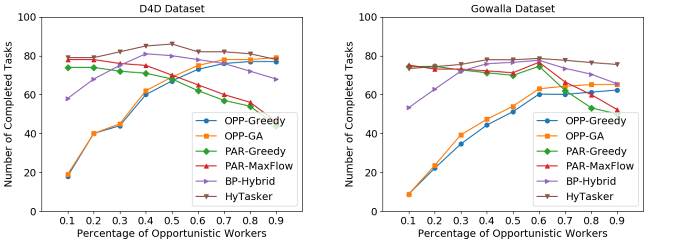


Fig. 9. Performance comparison under various percentage of opportunistic workers

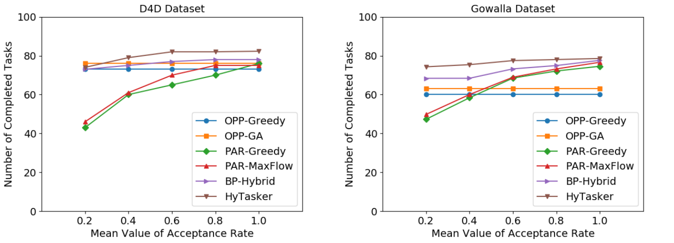


Fig. 10. Performance comparison under different values of μ (the mean value of task acceptance rate of PW)

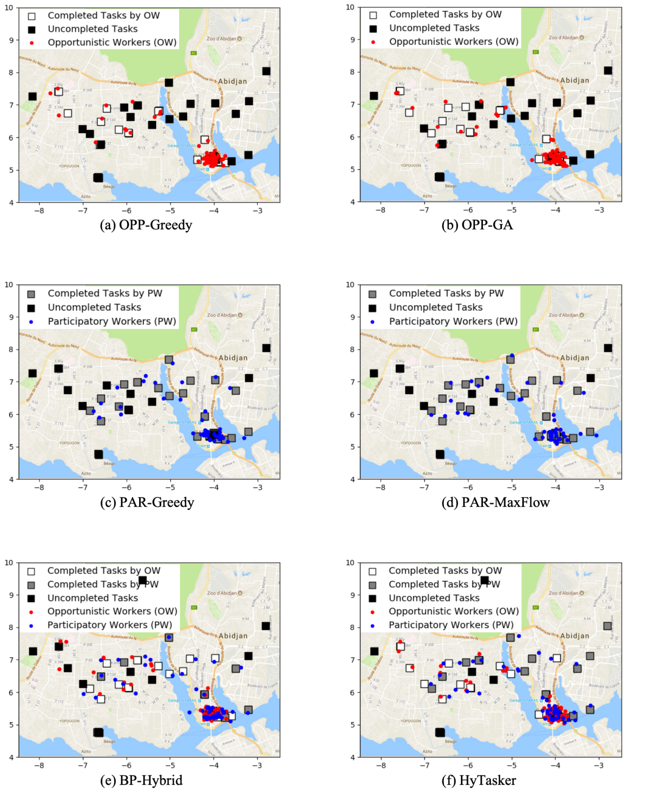
0.6. From Fig. 10, we can also see that HyTasker consistently outperforms other methods in all settings of μ and on both two datasets.

5.3.6 Detailed Analysis and Implications

The experimental results above show an overall comparison of different methods under various parameter settings. In this subsection, we will further present some details and corresponding analysis, which can verify some of our observations and intuitions. Due to the limited space, we only report the analysis result on D4D dataset.

Fig. 11 (a)-(f) visualize the distribution of tasks, historical locations of workers and final task completion status under a specific setting (i.e., number of workers is 300, number of tasks is 60, total budget is 800, and γ is 0.6) for D4D dataset. For the completed tasks achieved by BP-Hybrid, and HyTasker, we further use different legends to show whether they are completed by participatory workers or opportunistic workers.

From Fig. 11 (a) (b) (f), we can see that compared with opportunistic-mode-based approaches, the advantage of HyTasker is that it can complete some tasks in the worker-sparse areas, thus the number of completed tasks is increased. The comparison of Fig. 11(e) and Fig. 11(f) further demonstrates why HyTasker outperforms BP-Hybrid. From the distribution visualization of two types of workers, we can see that compared with BP-Hybrid, HyTasker can better leverage the opportunistic workers to complete tasks in areas where participatory workers are sparsely distributed. This indicates that: 1) the joint optimization of offline and online phase is more beneficial than considering them separately, and 2) it is effective to consider the density of participatory workers when selecting the opportunistic workers.



(a) OPP-Greedy

(b) OPP-GA

(c) PAR-Greedy

(d) PAR-MaxFlow

(e) BP-Hybrid

(f) HyTasker

Fig. 11. Worker distribution and task completion status for a single round: (a) OPP-Greedy; (b) OPP-GA; (c) PAR-Greedy; (d) PAR-MaxFlow; (e) BP-Hybrid; (f) HyTasker.

In addition, to test whether the use of location entropy is beneficial, we further compare HyTasker with two variants of HyTasker without using location entropy. One variant (called ”Equal-Weight”) sets equal weight for each task location, and the other variant (called ”Visiting Frequency as Weight”) sets the priority to be inversely proportional to the total number of past visits. Fig. 12 presents the average number of completed tasks among multiple rounds of experiments on D4D dataset when the total number of tasks is fixed at 90. We can see that with the adoption of location entropy, HyTasker can achieve more completed tasks than the two variants.

We further compare the performance between HyTasker and the optimal solution called OPT in a small search space. First, we significantly narrow the scale of sensing areas (only with 20 cell towers in D4D dataset and 20 subareas in Gowalla). Second, we set the total budget to be small to control the search space, which varies as 50, 75, 100, 125 and 150. OPT enumerates all possible sets of opportunistic worker. For each set, it then enumerates all possible combinations for task assignment regarding participatory workers. Fig. 13 shows the performance of HyTasker and OPT on two datasets, and HyTasker can averagely achieve 77.9% coverage compared to OPT. To some degree, this indicates that our approach achieves a relatively good approximation to the optimal solution of the original problem.

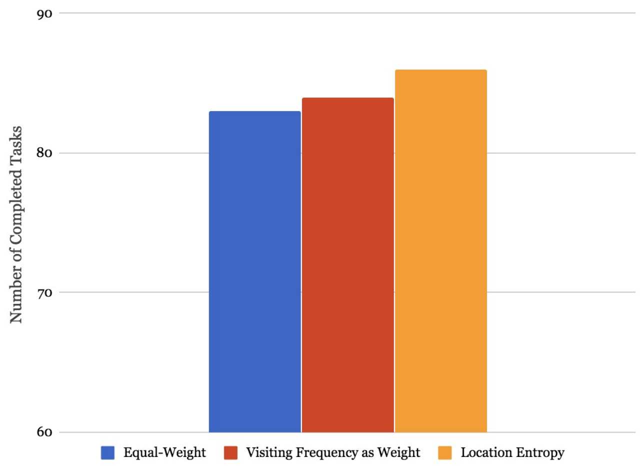


Fig. 12. Demonstrating the effectiveness of location entropy by comparing HyTasker with other two variants

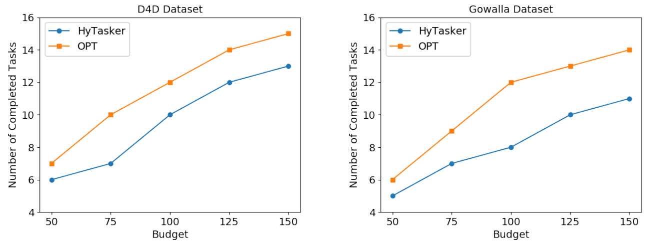


Fig. 13. Performance comparison in a small search space (left: D4D dataset; right: Gowalla dataset)

6 LIMITATION AND DISCUSSION

This section discusses other issues that are not addressed in this work due to space and time constraints, which we plan to investigate in our future work.

Task type and heterogeneity. Although the current implementation of HyTasker is for environmental sensing tasks, the idea of the hybrid task allocation can be extended to other types of tasks if supported by the mobility data. As an example, for traffic status monitoring tasks, the subareas are divided based on the road sections rather than cell towers. Theoretically, if we use more precise trajectory data (e.g., GPS readings) to model more fine-grained mobility pattern of users (e.g., within the granularity of road sections), HyTasker can support the traffic status monitoring tasks. However, there are several challenges when extending HyTasker to such type of tasks in real-world settings, which can be added into our future work. For example, fine-grained localization raises the concerns of energy consumption and privacy leaking. How to balance the mobility prediction accuracy and these concerns is a challenging research question. Besides, in the current version of HyTasker, we assume that multiple tasks are homogenous but distributed in different locations (e.g., with the same sensing duration). It would be useful to further study if HyTasker can be extended to an MCS platform with multiple heterogeneous sensing tasks.

Dynamic arrival of new tasks and workers. In this paper, we assume that all tasks have been pre-published before the worker selection phase, so that the number of tasks and the distribution of task locations are already known and fixed. However, for a multi-task-oriented MCS platform, new tasks may be published anytime online. Moreover, new workers may also come to the platform continuously. Thus, how to tackle the dynamic arrival of new tasks and workers is a key challenge, which is not addressed in this paper. This challenge could lead to new research issues, such as how to predict the dynamic arrival of new tasks and workers, which will be added as our future work to extend the functionality of HyTasker.

Timing for online task assignment. In the online phase of HyTasker, we assign a set of participatory workers and require them to move specifically to perform tasks that are not completed by the opportunistic workers. Here we assume that the participatory workers can complete the assigned tasks before the end time, and do not vary the timing of task assignment in this paper. In the experiments, we set this timing to one hour before the end of sensing period (e.g, 17:00 when the sensing period is 8:00-18:00). However, the best timing for online allocation still deserves further research in our future work. If the timing is too early, the participatory workers would complete more tasks which may be finished by opportunistic workers latter on, thus the total cost is higher. On the other hand, if the participatory workers are allocated too late, then they may not have enough time to travel and complete the tasks before the deadline.

Different incentive models. In HyTasker, the opportunistic workers get the same fixed reward for the entire sensing period, while the participatory workers get incentive rewards in proportion to their travel distances. In the MCS research community, there are actually a variety of incentive models [43], which are more complicated by considering multiple factors such as fairness, economic feasibility and data quality. In our future work, we plan to extend HyTasker by adopting more sophisticated incentive mechanisms (e.g., auction-based incentive models).

Learning of task acceptance rate. In this work, we assume that the task acceptance rate of each participatory worker has already been learned, which is simulated in the experiments [45], [46], [47]. To extend HyTasker to a wider range of application scenarios, we need to further improve it by learning and predicting the workers' task acceptance rate. For example, the authors in [47] presented a learning framework based on workers' previous participation history with the consideration of incentive reward and task distance. However, several challenges exist to achieve a good prediction accuracy, which can be the direction of our future work. First, factors affecting users' decisions is very complex [48] (e.g., task type, time availability, task distance, incentives, and even some emotional factors), and how to extract these features is non-trivial. Second, there are no historical participation records for new candidate workers, thus the prediction for their acceptance rate is challenging.

Spatial correlation among tasks. The goal of HyTasker is to maximize the number of completed tasks for a certain MCS campaign, in which we consider each task as equally important. Actually, the sensor readings in different subareas can be spatially correlated. For example, we can use the air quality information in one subarea to infer that of a nearby subarea. In our future work, we attempt to integrate mechanisms such as sparse crowd sensing [29] into HyTasker to further reduce the sensing cost.

More complex task quality metric. Actually, many factors could affect sensing task quality, including quality of workers, number of reports for each task, task type, availability of sensors, proximity of sensors to task locations, and so forth. Hence, establishing a sophisticated and practical task quality model is a very complex research issue itself, and several literatures such as [51] start to touch this issue but no perfect model exist yet. In the future work, we attempt to investigate if HyTasker can be extended with more sophisticated task quality metrics, which may result in a completely new research work.

7 CONCLUSION

In this paper, we proposed a two-phased hybrid MCS task allocation framework, called HyTasker. In the offline phase, HyTasker selects a group opportunistic workers and requires them to complete MCS tasks during their daily routines. In the online phase, HyTasker assigns another set of participatory workers and requires them to move specifically to perform tasks that are not yet completed. Since the two types of workers share a total budget, we proposed a greedy based opportunistic worker selection process by simultaneously considering the predicted task assignments for the participatory workers. Experiments on two real-world mobility datasets show that HyTasker outperforms other baseline methods.

8 ACKNOWLEDGEMENT

This work was supported by NSFC (National Natural Science Foundation of China) under Grant No. 61872010.

REFERENCES

[1] R. K. Ganti, F. Ye, and H. Lei. 2011. Mobile crowdsensing: Current state and future challenges. IEEE Communications Magazine, 49: 32-39.

[2] Guo, B., Wang, Z., Yu, Z., Wang, Y., Yen, N. Y., Huang, R., & Zhou, X. (2015). Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. ACM Computing Surveys (CSUR), 48(1), 7.

[3] M. Ra, B. Liu, T. F. L. Porta, and R. Govindan. 2012. Medusa: A programming framework for crowd-sensing applications. In Proceedings of the 10th international conference on Mobile systems, applications, and services (MobiSys '12), 337-350.

[4] Campaignr: http://research.cens.ucla.edu/urban/

[5] Taskrabbit. https://www.taskrabbit.com.

[6] Waze. https://www.waze.com

[7] V. Coric and M. Gruteser. 2013. Crowdsensing Maps of On-street Parking Spaces. In Proceedings of the 2013 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '13), 115-122.

[8] B. Guo, H. Chen, Z. Yu, X. Xie, S. Huangfu, and D. Zhang. 2015. FlierMeet: A Mobile Crowdsensing System for Cross-Space Public Information Reposting, Tagging, and Sharing. IEEE Transactions on Mobile Computing, 14: 2020-2033.

[9] Y. Zheng, F. Liu, and H. Hsieh. 2013. U-Air: When urban air quality inference meets big data. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '13), 1436-1444.

[10] Jiangtao WANG, Yasha WANG, Daqing ZHANG, Leye WANG, Chao CHEN, JaeWoong LEE, & Yuanduo, HE. (2017). Real-time and generic queue time estimation based on mobile crowdsensing. Frontiers of Computer Science, 11(1), 49-60.

[11] Ludwig, T., Reuter, C., Siebigteroth, T., & Pipek, V. (2015, April). CrowdMonitor: Mobile crowd sensing for assessing physical and digital activities of citizens during emergencies. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (pp. 4083-4092). ACM.

[12] Wang, J., Wang, L., Wang, Y., Zhang, D., & Kong, L. (2018). Task Allocation in Mobile Crowd Sensing: State of the Art and Future Opportunities. IEEE Internet of Things Journal, 2018, 5 (5), 3747-3757

[13] Kandappu, T., Jaiman, N., Tandriansyah, R., Misra, A., Cheng, S. F., Chen, C., ... & Dasgupta, K. (2016, September). Tasker: Behavioral insights via campus-based experimental mobile crowd-sourcing. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (pp. 392-402). ACM.

[14] Lu, Y., Xiang, S., Wu, W., & Wu, H. (2015, September). A queue analytics system for taxi service using mobile crowd sensing. In Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers (pp. 121-124). ACM.

[15] Liu, Y., Guo, B., Wang, Y., Wu, W., Yu, Z., & Zhang, D. (2016, September). Taskme: multi-task allocation in mobile crowd sensing. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (pp. 403-414). ACM.

[16] Ying, J. J. C., Lee, W. C., & Tseng, V. S. (2013). Mining geographic-temporal-semantic patterns in trajectories for location prediction. ACM Transactions on Intelligent Systems and Technology (TIST), 5(1), 2.

[17] To, H. (2016, October). Task assignment in spatial crowdsourcing: challenges and approaches. In Proceedings of the 3rd ACM SIGSPATIAL PhD Symposium (p. 1). ACM.

[18] Kandappu, T., Misra, A., & Tandriansyah, R. (2017, February). Collaboration Trumps Homophily in Urban Mobile Crowdsourcing. In Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (pp. 902-915). ACM.

[19] Kandappu, Thivya, et al. "Tasker: Behavioral insights via campus-based experimental mobile crowd-sourcing." Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, 2016.

[20] Kandappu, T., Misra, A., Cheng, S. F., Jaiman, N., Tandriansyah, R., Chen, C., ... & Dasgupta, K. (2016, February). Campus-scale mobile crowd-tasking: Deployment & behavioral insights. In Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (pp. 800-812). ACM.

[21] Sasank Reddy, Katie Shilton, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. Using context annotated mobility profiles to recruit data collectors in participatory sensing. In Location and Context Awareness, pages 52-69. Springer, 2009.

[22] Reddy, D. Estrin, and M. Srivastava. Recruitment framework for participatory sensing data collections. In Proceedings of Pervasive, pages 138-155. 2010.

[23] Giuseppe Cardone, Luca Foschini, Paolo Bellavista, Antonio Corradi, Cristian Borcea, Manoop Talasila, and Reza Curtmola. Fostering participaction in smart cities: a geo-social crowdsensing platform. Communications Magazine, IEEE, 51(6), 2013.

[24] Zhang, M., Yang, P., Tian, C., & Tang, S. (2015). Quality-aware sensing coverage in budget constrained mobile crowdsensing networks. IEEE Transactions on Vehicular Technology, 1-1.

[25] D Zhang, H Xiong, L Wang, and G Chen. Crowdrecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint. In Proc. ACM Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp), 2014.

[26] Guo, B., Liu, Y., Wu, W., Yu, Z., & Han, Q. (2017). Activecrowd: A framework for optimized multitask allocation in mobile crowdsensing systems. IEEE Transactions on Human-Machine Systems, 47(3), 392-403.

[27] Xiong, H., Zhang, D., Guo, Z., Chen, G., & Barnes, L. E. (2017). Near-Optimal Incentive Allocation for Piggyback Crowdsensing. IEEE Communications Magazine, 55(6), 120-125.

[28] Lane, N. D., Chon, Y., Zhou, L., Zhang, Y., Li, F., Kim, D., ... & Cha, H. (2013, November). Piggyback CrowdSensing (PCS): energy efficient crowdsourcing of mobile sensor data by exploiting smartphone app opportunities. In Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (p. 7). ACM.

[29] Wang, L., Zhang, D., Wang, Y., Chen, C., Han, X., & M'hamed, A. (2016). Sparse mobile crowdsensing: challenges and opportunities. IEEE Communications Magazine, 54(7), 161-167.

[30] Z. Song, C. H. Liu, J. Wu, J. Ma, and W. Wang. 2014. QoI-Aware Multitask-Oriented Dynamic Participant Selection with Budget Constraints. IEEE Transactions on Vehicular Technology, 63: 4618-4632.

[31] Wang, J., Wang, Y., Zhang, D., Xiong, H., Wang, L., & Sumi, H., et al. (2016). Fine-grained multi-task allocation for participatory sensing with a shared budget. Internet of Things Journal (in press).

[32] Wang, J., Wang, Y., Zhang, D., Wang, F., He, Y., & Ma, L. PSAllocator: Multi-Task Allocation for Participatory Sensing with Sensing Capability Constraints. The, ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW 2017).

[33] D. Deng, C. Shahabi, and U. Demiryurek. Maximizing the number of worker's self-selected tasks in spatial crowdsourcing. In SIGSPATIALGIS, 2013.

[34] L. Kazemi and C. Shahabi. Geocrowd: enabling query answering with spatial crowdsourcing. In SIGSPATIAL GIS, 2012.

[35] Kazemi, L., Shahabi, C., & Chen, L. (2013, November). Geotrucrowd: trustworthy query answering with spatial crowdsourcing. In Proceedings of the 21st sigspatial international conference on advances in geographic information systems (pp. 314-323). ACM.

[36] Cheng, P., Lian, X., Chen, L., & Shahabi, C. (2017, April). Prediction-Based Task Assignment in Spatial Crowdsourcing. In IEEE 33rd International Conference on Data Engineering, ICDE 2017 (pp. 997-1008). IEEE.

[37] P. Cheng, X. Lian, Z. Chen, and et al. Reliable diversity-based spatial crowdsourcing by moving workers. VLDB.

[38] Q. Liu, T. Abdessalem, H. Wu, Z. Yuan, and S. Bressan. Cost minimization and social fairness for spatial crowdsourcing tasks. In DASFAA, 2016.

[39] Mirzasoleiman, B., Badanidiyuru, A., Karbasi, A., Vondrk, J., & Krause, A. (2015, January). Lazier Than Lazy Greedy. In AAAI (pp. 1812-1818).

[40] V.D. Blondel, M. Esch, C. Chan, F. Clerot, P. Deville, E. Huens, F. Morlot, Z. Smoreda, and C. Ziemlicki. 2012. Data for development: the d4d challenge on mobile phone data. arXiv preprint arXiv:1210.0137.

[41] Cranshaw, J., Toch, E., Hong, J., Kittur, A., & Sadeh, N. (2010, September). Bridging the gap between physical location and online social networks. In Proceedings of the 12th ACM international conference on Ubiquitous computing (pp. 119-128). ACM.

[42] Reinelt, G. (1991). TSPLIB-A traveling salesman problem library. ORSA journal on computing, 3(4), 376-384.

[43] Jaimes, L. G., Vergara-Laurens, I. J., & Raij, A. (2015). A survey of incentive techniques for mobile crowd sensing. IEEE Internet of Things Journal, 2(5), 370-380.

[44] Li, Hanshang, Ting Li, and Yu Wang. ”Dynamic participant recruitment of mobile crowd sensing for heterogeneous sensing tasks.” Mobile Ad Hoc and Sensor Systems (MASS), 2015 IEEE 12th International Conference on. IEEE, 2015.

[45] Roy, S. B., Lykourentzou, I., Thirumuruganathan, S., Amer-Yahia, S., & Das, G. (2015). Task assignment optimization in knowledge-intensive crowdsourcing. The VLDB Journal, 24(4), 467-491.

[46] Zheng, Libin, and Lei Chen. ”Maximizing acceptance in rejection-aware spatial crowdsourcing.” IEEE Transactions on Knowledge and Data Engineering (2017).

[47] Karaliopoulos, M., Koutsopoulos, I., & Titsias, M. (2016, July). First learn then earn: Optimizing mobile crowdsensing campaigns through data-driven user profiling. In Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing (pp. 271-280). ACM.

[48] Gustarini, Mattia, Katarzyna Wac, and Anind K. Dey. ”Anonymous smartphone data collection: factors influencing the users' acceptance in mobile crowd sensing.” Personal and Ubiquitous Computing 20.1 (2016): 65-82.

[49] Lane, N. D., Eisenman, S. B., Musolesi, M., Miluzzo, E., & Campbell, A. T. (2008, February). Urban sensing systems: opportunistic or participatory?. In Proceedings of the 9th workshop on Mobile computing systems and applications (pp. 11-16). ACM

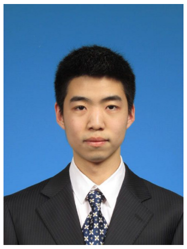
[50] Bowen Wang, Linghe Kong, Liang He, Fan Wu, Jiadi Yu, Guihai Chen. ”I(TS,CS): Detecting Faulty Location Data in Mobile Crowdsensing”. IEEE ICDCS, Vienna, Austria, 2018.

[51] Restuccia F, Ghosh N, Bhattacharjee S, et al. Quality of information in mobile crowdsensing: Survey and research challenges. ACM Transactions on Sensor Networks (TOSN), 2017, 13(4): 34.

[52] http://snap.stanford.edu/data/loc-Gowalla.html

[53] Wang J, Wang Y, Zhang D, et al. Multi-task allocation in mobile crowd sensing with individual task quality assurance. IEEE Transactions on Mobile Computing, 2018, 17 (9), 2101-2113.

Jiangtao Wang Jiangtao Wang received the PhD degree in Peking University, China, in 2015. He is currently a lecturer (assistant professor) in School of Computing and Communications at Lancaster University, UK. Before that, he was an assistant professor at Peking University. His research interest includes mobile and pervasive computing, crowd sensing, and crowdsourcing.



Feng Wang Feng Wang is an undergraduate student at School of Electronic Engineering and Computer Science, Peking University, China. His research interest is mobile crowd sensing.

Yasha Wang Yasha Wang received his Ph.D. degree in Northeastern University, Shenyang, China, in 2003. He is a professor and associate director of National Research & Engineering Center of Software Engineering in Peking University, China. His research interest includes urban data analytics, ubiquitous computing, software reuse, and online software development environment. He has published more than 50 papers in prestigious conferences and journals, such as ICWS, UbiComp, ICSP and etc. As a technical leader and manager, he has accomplished several key national projects on software engineering and smart cities. Cooperating with major smart-city solution providing companies, his research work has been adopted in more than 20 cities in China.

Leye Wang Leye Wang obtained his Ph.D. from Institut Mines-Télécom/Télécom SudParis and Université Pierre et Marie Curie, France, in 2016. He received his M.Sc. and B.Sc. in computer science from Peking university, China. He is currently an assistant professor in Institude of Software, School of Electronic Engineering and Computer Science. His research interests include mobile crowdsensing, social networks, and intelligent transportation systems.

Zhaopeng Qiu Zhaopeng Qiu is an undergraduate student at School of Electronic Engineering and Computer Science, Peking University, China. His research interest is data analysis and NLP.

Daqing Zhang Daqing Zhang is a professor at Peking University, China, and Télécom Sud-Paris, France. He obtained his Ph.D from the University of Rome ”La Sapienza,” Italy, in 1996. His research interests include context-aware computing, urban computing, mobile computing, and so on. He served as the General or Program Chair for more than 10 international conferences. He is an Associate Editor forACM Transactions on Intelligent Systems and Technology,IEEE Transactions on Big Data, and others.

Bin Guo Bin Guo received the PhD degree in computer science from Keio University, Japan, in 2009, and then was a postdoc researcher with Institut Télécom SudParis, France. He is a professor with Northwestern Polytechnical University, China. His research interests include ubiquitous computing, mobile crowd sensing, and human-computer interaction.

Qin Lv Qin Lv received her PhD degree in computer science from Princeton University in 2006. She is an associate professor in the Department of Computer Science, University of Colorado Boulder. Her main research interests are data-driven scientific discovery and ubiquitous computing. Her research spans the areas of multi-modal data fusion, spatial-temporal data analysis, anomaly detection, mobile computing, social networks, recommender systems, and data management. Her research is interdisciplinary in nature and interacts closely with a variety of application domains including environmental science, Earth sciences, renewable and sustainable energy, materials science, as well as the information needs in people’s daily lives. Lv is an associate editor of ACM IMWUT, and has served on the technical program committee and organizing committee of many conferences. Her work has more than 4,000 citation