

# Task Allocation in Mobile Crowd Sensing: State-of-the-Art and Future Opportunities

Jiangtao Wang<sup>1</sup>, Leye Wang, Yasha Wang, Daqing Zhang, and Linghe Kong<sup>2</sup>

**Abstract**—Mobile crowd sensing (MCS) is the special case of crowdsourcing, which leverages the smartphones with various embedded sensors and user’s mobility to sense diverse phenomenon in a city. Task allocation is a fundamental research issue in MCS, which is crucial for the efficiency and effectiveness of MCS applications. In this paper, we specifically focus on the task allocation in MCS systems. We first present the unique features of MCS allocation compared to generic crowdsourcing, and then provide a comprehensive review for diversifying problem formulation and allocation algorithms together with future research opportunities.

**Index Terms**—Crowdsourcing, mobile crowd sensing (MCS), task allocation.

## I. INTRODUCTION

URBAN sensing is crucial for understanding the current status of a city in many aspects (e.g., air quality, traffic status, noise level, etc.). With the development of Internet of Things (IoT), mobile Internet and cloud computing, we now have various ways to collect urban information [1]–[3]. Among them, the prevalence of mobile devices and the increasing smart sensing requirements in the city have led to an alternative or complementary approach for urban sensing, called mobile crowd sensing (MCS) [4]. Similar concepts include participatory sensing [5], location-based/mobile/spatial crowdsourcing [6], collaborative sensing [7], and so forth.

MCS leverages the inherent mobility of mobile users (i.e., participants or workers), the sensors embedded in mobile phones and the existing communication infrastructure (Wi-Fi and 4G/5G networks) to collect and transfer urban sensing data. MCS has enabled diverse applications, such as air quality monitoring [33], noise level sensing [18], queue time estimation [66], risky mountain trail detection [76], and so forth. Compared to wireless sensor networks, which are based on specialized sensing infrastructures, MCS is less costly and

can obtain a higher spatial–temporal coverage. As a result, the emergence of MCS has expanded the scope of IoT, where the “things” are not only limited to physical objects (i.e., they also include human and their carried mobile devices).

The connection between tasks and workers is crucial for the success of MCS applications. The simplest way is that the organizers publish various MCS tasks and workers select tasks themselves based on their location and preferences (e.g., Medusa [9] and PRISM [10]), which is called the pull mode. The pull mode is easy to implement. However, for the pull mode, the cloud server does not have any control over the tasks assignments. Since workers select tasks based on their own preference or goals (e.g., nearby, easy, or high payment), the overall performance may not be globally optimized. For example, some sensing tasks have few participants so that the sensing quality is low, while others may have too many which leads to redundant sensing data.

Therefore, it is a promising technical alternative that the server automatically assigns sensing tasks to workers according to the system optimization goals (e.g., maximizing the sensing quality while ensuring the budget constraints), which is called the push mode. In recent years, the studies for automatic MCS task allocation becomes a hotspot in research communities, such as ubiquitous computing, social computing, cooperative computing, and computer network.

There are some tutorials or surveys (e.g., [1] and [24]) for MCS in recent years. The scope of these papers is for the entire research community of MCS, which discuss different aspects and research issues in this field to give us an overview picture of MCS. However, as these survey papers mainly focus on the general and overall research picture and roadmap of MCS, none of them summarize and discuss the research problem of sensing task allocation in details and systematically. Especially as task allocation is one of the hottest research topics in MCS where there are still continuous achievements published in top venues across various areas in recent years (e.g., ICDE [71], UbiComp [32], CSCW [50], WWW[64], IEEE TMC [72], and IEEE TIST [73]), a tutorial or survey devoted to summarizing its up-to-date research results is even desirable. To this end, in this paper, we specifically focus on the task allocation problem in MCS and provide a comprehensive review with future research opportunity.

The possible inspiration derived from this paper consists of the following aspects.

- 1) We analyze the unique factors or features in MCS in addition to general crowdsourcing, which can reveal

Manuscript received October 9, 2017; revised April 3, 2018 and June 24, 2018; accepted August 4, 2018. Date of publication August 8, 2018; date of current version November 14, 2018. This work was mainly funded by NSFC Grant (No. 61702017). (Corresponding authors: Jiangtao Wang; Yasha Wang.)

J. Wang, Y. Wang, and D. Zhang are with the Key Laboratory of High Confidence Software Technologies, Ministry of Education, Beijing 100871, China.

L. Wang is with the Computer Science and Engineering Department, Hong Kong University of Science and Technology, Hong Kong (e-mail: wly@cse.ust.hk).

L. Kong is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China.

Digital Object Identifier 10.1109/JIOT.2018.2864341



Fig. 1. Crowdsourcing used in different domains.

why the traditional task assignment methods for crowdsourcing cannot be directly utilized to tackle the task allocation problem in MCS.

- 2) We present and summarize different types of problem formulation in MCS task allocation and corresponding algorithms, which help the researchers or engineers to quickly identify the subset of studies and provide guidance or inspiration when designing and implementing the MCS systems or applications.
- 3) We discuss some potential research directions and proposals, which aims to consider more practical issues in MCS task allocation.

## II. PRELIMINARY FOR MCS

### A. Crowdsourcing and MCS

The term “crowdsourcing” was coined by Howe [11] to describe how businesses were using the Internet to outsource work to the crowd. The basic idea of crowdsourcing is to leverage the power of crowd to collaboratively complete a complex task, where each individual (called “worker”) only completes much easier micro-tasks. In recent years, crowdsourcing-based systems are widely used in many domains [12]–[16] (see Fig. 1), and the intersection between crowdsourcing and these tradition research areas gives rise to a new research topic.

The popularity of mobile devices and the increasing sensing requirement in the city enable a subclass of crowdsourcing called the MCS [1]. Similar to the notion of participatory sensing [5] and human-centric computing [17], MCS refers to the sensing paradigm in which users with sensor-rich mobile devices collect and contribute data in order to enable various applications. In an MCS system, there are two key players, i.e., workers (or participants) who collect and report sensing data through mobile device, and task organizers who manage and coordinate whole MCS process. Various kinds of MCS applications have been proposed and implemented in both academic and industry areas, such as environmental applications [18], [19], infrastructure applications [8], [20], [21], and social applications [22], [23]. For detailed introduction and classification about various MCS applications, interested readers can refer to a recent survey paper [29].

### B. Life-Cycle of MCS and Research Issues

The life-cycle of MCS can be divided into four stages: 1) task creation; 2) task allocation; 3) task execution; and 4) data aggregation. The main functionality and research issues of each stage are briefly described as follows.

- 1) *Task Creation*: The MCS organizer creates an MCS task through providing the workers with the corresponding mobile phone applications. In this stage, the key research issue is how to improve the efficiency of MCS task creation, especially for those who do not have professional programming skills [9], [25].
- 2) *Task Allocation*: After the organizer creates an MCS task, the next stage is task allocation, in which the application or a public platform recruits workers and assigns them with sensing tasks. The key research issue at this stage is how to optimize the task allocation with the consideration of diverse factors, such as spatial coverage, incentive cost, energy consumption, and task completion time [46], [50].
- 3) *Task Execution*: Once receiving the assigned micro sensing tasks, the workers complete them within a predefined spatial-temporal scale (i.e., time duration and target region). This state includes sensing, computing, and data uploading. How to save energy consumption is the major research issue in this stage [27], [28].
- 4) *Crowd Data Integration*: This stage aggregates the reported data from the crowd according to the requirement of task organizers. The key issue in this stage is how to infer missing data and provide a complete spatial-temporal picture of the target phenomenon (e.g., the real-time air quality map in the city) [39], [40].

In this paper, we specifically focus on the task allocation stage. We first present the unique features of MCS allocation compared to generic crowdsourcing. Then, we provide a comprehensive review for diversifying problem formulation and allocation algorithms together with future research opportunities.

## III. SPECIFIC FACTORS IN MCS TASK ALLOCATION

### A. Overview

MCS is the special case where the idea of crowdsourcing is used in urban sensing scenarios. Task allocation of MCS shares some common concerns or factors with general crowdsourcing tasks (e.g., article writing or image classification) [30]. For example, both general crowdsourcing and MCS consider incentive models and budget constraints in task allocation strategies. On the other hand, MCS has its own unique features which differ from general crowdsourcing. To this end, we provide the comparative schemas of general crowdsourcing and MCS in Fig. 2, where the green color labels the unique factors of MCS.

Essentially, the unique characteristic of MCS lies in the aspects of *mobility* and *sensing*. Thus, we elaborate the MCS-specific factors or features from these two aspects.

1) *Mobility-Relevant Features*: Different from general crowdsourcing tasks, MCS requires the workers to complete sensing tasks in certain locations, because the sensing results

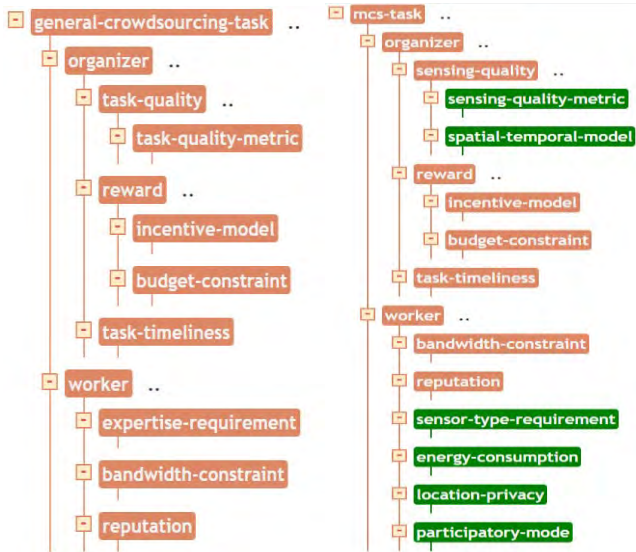


Fig. 2. Comparative schemas of general crowdsourcing and MCS (left: general crowdsourcing and right: MCS. Green one labels the unique factors for MCS compared to general crowdsourcing).

are location-dependent (e.g., air quality, noise level, and traffic congestion status). This characteristic leads to the “participatory mode” and “location privacy” features in Fig. 2. First, based on how the workers move to the locations for sensing, we can divide MCS task allocation into two participation modes (i.e., participatory or opportunistic). Second, since MCS usually targets at collecting *spatial* data all across a city, location privacy should be carefully preserved. In addition, spatial-temporal models usually need to be considered in the sensing quality metric of MCS, but rarely in the task quality of general crowdsourcing.

2) *Sensing-Relevant Features*: Different from general crowdsourcing, MCS always targets at urban sensing tasks. First, the execution of sensors and localization modules introduces much more energy consumption into MCS than general crowdsourcing. The energy consumption has a direct impact on the battery life of a worker’s smartphone. If the energy consumption of an MCS task is too high, it will severely reduce the mobile phone users’ willingness of becoming a crowd worker. Therefore, it is important to control the energy consumption of workers in the MCS systems, which is also labeled as a unique feature in Fig. 2. Second, many MCS tasks need to invoke phone-embedded sensors for task completion, but the set of sensors for each worker may be different as they hold various brands and models of smart devices. Thus, the “sensor type requirement” should be particularly considered in the task allocation of MCS.

As the worker and organizer are the key roles in MCS, we divide the above MCS-specific factors into two categories from the perspective of worker and organizer, respectively.

## B. Worker-Side Factors

### 1) Workers’ Participation Mode:

- *Participatory Mode*: This mode requires the workers to change their original routes and specifically move to certain places to complete MCS tasks [31], [32],

and its advantage is that it 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.

- *Opportunistic Mode*: For this mode, workers can complete tasks unintentionally during their daily routines without the need to change their routes [33], [34]. 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.

2) *Location Privacy*: There have been proposed a spectrum of location privacy preserving techniques for location-based services, and many of them have also been successfully adopted in MCS [65]. Different privacy mechanisms may have their own metrics to quantify the privacy protection effect. For example, the *cloaking* mechanism is often designed based on the *k-anonymity* metric, i.e., ensuring that a user’s reported location is same as the other  $k - 1$  users (i.e., a user is indistinguishable from the other  $k - 1$  users) [68];  $\epsilon$ -differential-privacy is a location obfuscation scheme to protect users’ real locations, which is able to bound the adversary’s posterior knowledge improvement over his prior knowledge about a user’s location, while  $\epsilon$  can be set by users’ privacy preferences [64]. In other words, if an adversary foreknows that a user has a probability of  $P$  in a location  $L$ , with the  $\epsilon$ -differential-privacy protection, the adversary’s confidence probability of the user at  $L$  will not be larger than  $C * P$  after observing the user’s obfuscated location, where  $C$  is a constant determined by  $\epsilon$ . As location privacy protection mechanisms generally include noises added into participants’ locations, it will bring novel challenges for task allocation, e.g., locations of users’ uploaded data become somehow uncertain [69] and the distance between users and task locations cannot be precisely measured [64]. Then, finding the optimal privacy mechanism, where the loss of task allocation efficiency is minimized, becomes rather important.

3) *Energy Consumption*: Several methods proposed in for mobile phone sensing [35] can be directly used to reduce the energy consumption for an individual worker, which are mainly adopted in the sensing and data uploading phase of MCS. Additionally, we can further optimize the overall energy consumption by designing more sophisticated task allocation mechanism [26]. In this paper, we focus on how to take the energy consumption concern into consideration in the task allocation phase.

## C. Organizer-Side Factors

1) *Spatial-Temporal Model*: Different from general crowdsourcing, task organizers in MCS can obtain a spatial-temporal



overview of the environment in the target area (e.g., air quality map in Beijing) by collecting sensor readings from mobile users. The most common way of modeling the time and space in MCS is to divide the entire sensing areas and time period into some equal-size subareas (1 km×1 km) and equal-length time slots (1 h per slot), so that we can get a number of spatial-temporal cells [33], [34], [49], [50]. Another way is regarding the sensing target as a point-of-interest (POI) with a given range (e.g., a circle with 100 m radius). If a mobile user moves inside such a range of a POI, he/she can collect the sensing data at this point [53]–[56]. Most of the general crowdsourcing tasks do not consider the location of the workers and sensing cycles. But for some, such as Internet quality measurement [57], the time and location of the reported network quality information is also considered. However, their spatial-temporal models are quite different, where the topology (the spatial model) and peak hours (the temporal model) of the network are considered to measure the service quality.

2) *Sensing Quality*: For MCS task allocation, the quality of sensing data is a primary concern for the task organizer. Thus, how to model or quantify the quality of sensing task in MCS should be considered. The sensing quality metric can be divided into the following two types.

- *Spatial-temporal coverage-based metrics*. One naive to measure the quality of sensing data is based on the number of collected data samples. Accordingly, a common metric to measure the sensing quality of an MCS tasks is the spatial-temporal coverage, i.e., how many subareas can be covered by the sensing data collected [33], [34], [49], [50]. It is also different in defining whether a subarea is “covered” or not. To simplify the problem, earlier research works always assume that if one subarea gets one data sample, it is regarded as covered in this time slot. However, recent studies such as [50] assume that at least a number of samples is needed (i.e., the minimum threshold) to guarantee the reliability of collected data. Then, if the minimum requirement is met, the coverage quality would increase as the number of samples increases until reaching to a certain degree (i.e., the maximum threshold).
- *Sensor data value-based metrics*. Due to the temporal and spatial correlations in the MCS systems, the sensor readings of some spatial-temporal cells can be inferred from the others. In this case, another typical way to quantify the sensing quality is to infer the data of subareas without sensor readings and then compute the inference error [39]–[41]. Especially, the average inference error among all the subareas is also often used as a quality metric [39], [40] for continuous sensing values (e.g., temperature), while average classification error is used for classification-based sensing values (e.g., air quality level) [41].

#### IV. PROBLEM FORMULATION IN MCS TASK ALLOCATION

Task allocation for MCS tasks is commonly formulated as the mathematical optimization problems with various goals

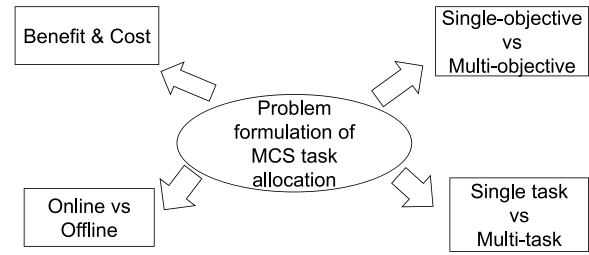


Fig. 3. Classification of problem formulation in MCS task allocation.

and constraints. We classify and summarize the state-of-the-art research works from the following perspectives (see Fig. 3).

##### A. *Benefit and Cost*

The essence of MCS task allocation is to achieve the trade-off between several opposing factors, which is divided into two classes called *benefit* and *cost*. The benefit is defined as the sensing quality of an MCS task, which may be measured by different metrics (i.e., spatial-temporal coverage-based metrics and sensor data value-based metrics described in Section III-C). However, in order to achieve higher benefit (or sensing quality), some overhead factors, which is called *cost* is this paper, should be taken into account. The cost factors mainly include incentive cost, risk of location privacy leak, energy consumption, and so forth. We classify the existing work as follows, which is based on the type of cost in the formulated optimization problems.

First, to improve the sensing quality of MCS tasks, the naïve way is to assign tasks to as many workers as possible. However, too many task assignments will lead to the increase in incentive cost. Thus, sensing quality and cost are two opposing factors, and managing the tradeoff between them through task allocation is a fundamental and crucial research problem. Several research studies were proposed recently which aimed at either maximizing the sensing quality with budget constraints (e.g., [42]–[44]), or minimizing the incentive cost while guaranteeing a minimum level of sensing quality (e.g., [33], [45], and [46]).

Second, sensing quality and location privacy are often two conflicting-objectives in task allocation optimization. To protect mobile users’ locations, their actual locations are often perturbed or obfuscated before being uploaded to the server. Usually, the higher protection effect is desired (i.e., the location is more inaccurate), the lower sensing quality could be obtained. One of the most commonly used location privacy protection methods belongs to the category of cloaking, where a user’s fine-grained location is down-graded to a coarse-level region [65]. More recently, differential privacy is applied in MCS to provide a theoretical privacy guarantee regardless of any adversary’s prior knowledge about his victim user’s location distribution [64]. To obtain the highest sensing quality while ensuring privacy protection effect, many researchers have formulated an optimization problem for task allocation, where privacy protection effect is often regarded as constraints and sensing quality as optimization objectives [64], [66].

Third, several methods can be utilized to reduce the energy consumption in MCS. For example, in sensing data collection phase, Khan *et al.* [35] designed new methods using a set of energy-efficient sensors to replace the traditional approaches consisting of more energy consuming sensors, or dynamically adjust the data collection frequency to do tasks more efficiently. In the data transferring phase, low-power wireless communication network (e.g., Wi-Fi) is utilized to upload data, rather through 3G/4G [47], or upload data to the server when users established the Internet connections for other applications, called piggyback [27]. However, all the above mechanisms are used for reducing the energy consumption of an individual worker. In the task allocation, several studies focus on how to optimize the overall energy consumption for MCS systems. For instance, to minimize the energy consumption, [33] attempts to minimize total energy consumption while ensuring the required spatial-temporal coverage. Liu *et al.* [36] formulated a task allocation problem, whose objective is to maximize sensing quality while minimizing energy consumption. Xiong *et al.* [34] formulated another MCS task allocation problem, in which the objective is to maximize the task quality given a limited overall energy consumption. The study in [51] formulates another version of task allocation problem by considering the energy consumption, worker's reputation, and budget limitation all together.

#### B. Single-Objective Allocation Versus Multiobjective Allocation

Most of the existing research works formulate the MCS task allocation as a single objective optimization problem, in which they only aim at optimizing one specific goal while keeping others as constraints. For example, the formulated problems in [33], [34], and [42]–[45] are all single-objective-oriented. On the other hand, some others formulate the MCS task allocation as a multiobjective optimization problem [32], [36]. For example, [36] aims at maximizing sensing quality while minimizing energy consumption in MCS task allocation. The objective of [32] is to minimize the traveling cost and meanwhile maximizing the number of completed MCS tasks. The multiobjective optimization problems formulated in [32] and [36] are commonly transformed into single-objective optimization problems based on the theory in [37], in which the weight of each objective is defined by task organizer. The shortcoming of such transformation is that sometimes it is difficult for the task organizers to decide the weight parameters.

#### C. Single-Task-Oriented Allocation Versus Multitask-Oriented Allocation

In the earlier stage of MCS research, existing approaches (e.g., [33], [34], [36], [39], [42], and [43]) are mostly single-task oriented, where they assume that tasks on MCS platforms are isolated so that the task allocation is executed for each single task independently. However, as the number of MCS tasks increases, the tasks are no longer independent, because they compete with each other in a shared and limited resource pool (e.g., shared user pool or total budget). Thus, in order

to better coordinate tasks and make full use of the limited resources, some recent studies (e.g., [48]–[50] and [77]) have started to focus on multitask allocation, where the interdependency of multiple tasks is considered. Typically, the objective is to optimize the overall utility of multiple tasks. For example, [50] and [77] studied the overall utility maximization of multiple tasks with worker's sensing capability constraints, while [32], [48], and [49] proposed frameworks to optimize the overall utility with a total incentive budget constraint. In these works, the overall utility is all defined as the weighted sum of each task's sensing quality (e.g., spatial-temporal coverage).

#### D. Offline Allocation Versus Online Allocation

In terms of the timing when the allocation solution is determined, MCS tasks allocation can be either online or offline. If the tasks are assigned before the start time of the MCS task execution, it is the offline mode. On the contrary, if the task allocation is performed while the MCS task is running, it is the online allocation. For example, studies such as [33], [34], [42], [49], and [50] are based on offline mode. The offline mode does not require the workers' real-time location information, which is more privacy-preserving. However, one main technical challenge for offline task allocation is that the system should be able to predict the workers' mobility accurately based on historical records. In contrast, existing studies, such as [60] and [61], adopt the online mode. The objective of [60] is to minimize the number of redundant task assignments while ensuring the required number of participants returning the sensing results within each time slot. A study in [61] aims at minimizing the number of assigned tasks while ensuring the full coverage the target area in each time slot. Compared to the offline mode, online task allocation has more knowledge about the real-time location of worker  $u$  in time slot  $i$ , if  $u$  uploads data with geotagging in previous time slots  $(1, 2, \dots, i - 1)$ . Thus, the mobility prediction can be easier with the combination of both real-time location and historical mobility records.

### V. MCS TASK ALLOCATION ALGORITHMS

#### A. General Framework

Though with different goals and constraints, the task allocation can be formulated as combinatorial optimization problems, which attempt to find an optimal solution from a large search space. For instance, several studies aim to find a subset set of workers [33], [34], [46], [48], while some others' goal is to find a subset of task-and-worker pairs [31], [32], [49], [50], [53], [54]. Intuitively, it is easy to think of a brute-force approach, where it can estimate the utility of each possible combination so that the optimal one can be obtained. However, the formulated combinatorial optimization problems are usually NP-hard, thus the brute force approach is not acceptable when there are a large number of workers or tasks. Therefore, existing research work commonly chooses to design approximation allocation algorithms to achieve the near-optimal solution, which can be divided into the following two categories.

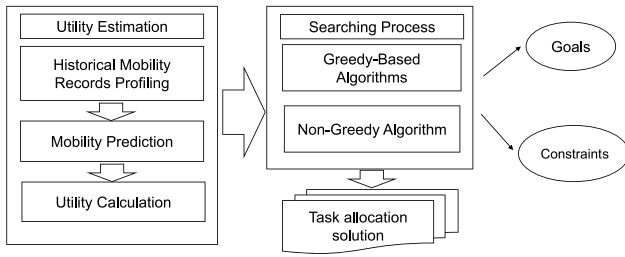


Fig. 4. General framework for MCS task allocation.

The general framework for MCS task allocation is presented in Fig. 4, which consists of two major components.

- 1) *Utility Estimation*: The algorithms for estimating the utility of a given set (a set of workers or task-and-worker pairs). Usually, the estimation needs the understanding of the workers' mobility pattern so that the historical mobility records profiling and mobility prediction are the basic components.
- 2) *Searching Process*: The searching algorithms to obtain a near-optimal solution. The algorithms are divided as greedy or nongreedy in this paper.

### B. Greedy-Based Algorithms

Most of the existing studies for MCS task allocation adopt the greedy-based algorithms [33], [34], [42], [43], [48]–[50], in which it iteratively selects “best” element (i.e., a worker or a task-and-worker pair) and adds into a set until certain stopping criterion is triggered (e.g., budget is used up, required coverage is reached, or none of the workers can be assigned to new tasks). After the greedy process stops, the obtained set is the near-optimal solution. For example, the greedy-based algorithm in [33] iteratively selects the participant with the maximum estimated coverage increase until the coverage requirement is satisfied. In some special problem settings such as [34] and [50], as the utility function cannot be estimated directly, multiple rounds of the greedy process should be executed to improve the optimality. In the experimental evaluation, the proposed greedy-based approaches are proved to be effective in their targeted application scenarios under various settings.

However, in terms of theoretical bound, some algorithms have approximation bound guarantees, while others do not. Whether the proposed greedy algorithms have approximation bound are determined by the property of the defined utility functions and constraints. For example, the utility functions defined in [33], [34], [42], [43], [48], and [49] are the submodular set function with cardinality constraints, so the greedy-based approaches can achieve at least  $1 - 1/e (\approx 0.63)$  approximation bound compared to the optimal solution. We look at the exemplary problem, which is defined as selecting a fixed number of users with the objective of maximizing coverage. In this example, with a limited number of workers (e.g., 1000 workers), if the optimal solution (e.g., the brute-force approach) can get a coverage of 0.98, then the near-optimal solution can get at least a coverage of  $0.9 \times 0.63 = 0.617$  even in the worst case. On the contrary, the utility functions used in [31], [32], and [50] are nonsubmodular, so that the

strict approximation bound is not declared in these studies. Considering that greedy-based algorithms enjoy a good empirical performance in studies such as [50], in which the utility functions are nonsubmodular, it is interesting to investigate if a certain theoretical bound exists in future work. Recent literature regarding the guarantees for greedy optimization of nonsubmodular functions [52] may inspire us to address this issue.

### C. Nongreedy Algorithms

Although through empirical studies the greedy-based approaches are proved to be effective in their formulated problems and settings, they are not the skeleton key for all rooms as the formulation of MCS task allocation is diversifying. The greedy algorithms are suboptimal in some scenario because they select the local best at each step. Therefore, more sophisticated algorithms have been designed.

For example, genetic algorithms (GAs) are used in [31] for optimizing time-sensitive and time-tolerant MCS task allocation problems. In GA, through several generations of selection, crossover, and mutation, the initial population (i.e., initial task allocation solution) converges to the optimal or near-optimal solution. Liu *et al.* [32] transformed the problem using the minimum cost maximum flow (MCMF) theory and construct a new MCMF model by considering different constraints, then propose the MT-MCMF and MTP-MCMF algorithms. Both [53] and [54] formulate the MCS task allocation as a bipartite graph partition problem and propose divide-and-conquer algorithms. In order to maximize the number of tasks allocated to each worker, two algorithms are developed using dynamic programming and branch-and-bound strategies [55]. Also, a bisection-based algorithm<sup>1</sup> is developed in [56] that performs top-down recursive bisection and a bottom-up merge procedure iteratively so that assignment and scheduling can be performed locally in a much smaller promising space.

### D. Algorithm Evaluation

We should evaluate the task allocation algorithm before applying it in real-world MCS systems. The common strategy for evaluating the algorithms is to compare the performance with different baselines under various settings (e.g., the number of tasks and workers, workers' bandwidth, total incentive budget, task distribution, etc.). One of the biggest challenges for the MCS research community to evaluate the task allocation algorithm is the absence of public real-world datasets from applications. Therefore, the existing work always evaluates the algorithms' performance based on both the real-world and synthetic datasets. The information of workers' mobility is usually based on a real-world dataset (such as D4D [2] and Gowalla [70]), while the information of task (such as spatial-temporal distribution, budget, and required quality) are commonly synthetic. A typical example of the real-world dataset used is the D4D dataset [2], which contains two data types. One data type contains the information about cell towers, including tower id, latitude, and longitude.

<sup>1</sup>The bisection method is a root-finding method that repeatedly bisects an interval and then selects a subinterval in which a root must lie for further processing.

TABLE I  
SUMMARY OF CHARACTERISTICS OF PROBLEM FORMULATION AND ALLOCATION ALGORITHM FOR EACH ANALYZED PAPER

| reference | sensing quality metric          | type of cost       | single/ multi objective | single/multi task | online /offline | algorithm                  |
|-----------|---------------------------------|--------------------|-------------------------|-------------------|-----------------|----------------------------|
| 31        | spatial-temporal coverage based | incentive cost     | single                  | multiple          | online          | Genetic Algorithms         |
| 53        | spatial-temporal coverage based | incentive cost     | single                  | multiple          | online          | divide-and-conquer         |
| 55        | spatial-temporal coverage based | incentive cost     | single                  | multiple          | online          | branch-and-bound           |
| 32        | spatial-temporal coverage based | incentive cost     | multiple                | multiple          | online          | MCMF                       |
| 54        | spatial-temporal coverage based | incentive cost     | multiple                | multiple          | online          | divide-and-conquer         |
| 56        | spatial-temporal coverage based | incentive cost     | multiple                | multiple          | online          | Bisection-based            |
| 33,42,45  | spatial-temporal coverage based | incentive cost     | single                  | single            | offline         | greedy                     |
| 34        | spatial-temporal coverage based | energy consumption | single                  | single            | offline         | greedy                     |
| 36,47     | spatial-temporal coverage based | energy consumption | multiple                | single            | offline         | greedy                     |
| 39        | sensor data value based         | incentive cost     | multiple                | multiple          | online          | matrix completion          |
| 43        | sensor data value based         | incentive cost     | single                  | multiple          | online          | greedy                     |
| 44        | spatial-temporal coverage based | incentive cost     | single                  | multiple          | online          | rule-based                 |
| 46        | spatial-temporal coverage based | incentive cost     | multiple                | single            | offline         | probabilistic registration |
| 27        | spatial-temporal coverage based | energy consumption | single                  | multiple          | online          | dynamic programming        |
| 48,49,50  | spatial-temporal coverage based | incentive cost     | single                  | single            | offline         | greedy                     |
| 51        | spatial-temporal coverage based | energy consumption | multiple                | multiple          | online          | MMO                        |
| 60        | spatial-temporal coverage based | energy consumption | multiple                | multiple          | online          | adaptive pace control      |
| 61        | spatial-temporal coverage based | energy consumption | single                  | multiple          | online          | adaptive pace control      |
| 64        | sensor data value based         | location privacy   | single                  | multiple          | online          | non-linear programming     |
| 66        | sensor data value based         | location privacy   | multiple                | multiple          | online          | probabilistic registration |

The other one contains 50 000 users' phone call records in Ivory Coast. The D4D dataset is used in the evaluation of task allocation algorithms such as [32]–[34], [49], and [50]. For the synthetic dataset, one representative example is that To *et al.* [59] proposed a toolbox to generate synthetic data for experimentation of MCS, thus leading to reproducible research.

Table I summarizes the characteristics of problem formulation and allocation algorithm for each MCS task allocation study. We hope this could help readers quickly identify the subset of relevant papers for his/her purposes.

## VI. FUTURE RESEARCH OPPORTUNITIES AND PROPOSALS

Existing work on MCS has studied various aspects for task allocation. However, the gap between ideal problem setting and real-world applications still prevent MCS system from being widely deployed. Thus, we next highlight several directions for future research by taking some practical issues into account.

### A. Sustainable MCS Task Allocation

Existing studies usually focus on short-term task allocation in MCS. For instance, the organizer allocates the sensing task



of traffic accident detection to participants immediately and the participants attempt to complete the tasks as soon as possible. In contrast, there are also many long-term sensing tasks, such as air quality surveillance for several years, which are significant for future cities and do need the sustainable task allocation. To achieve the sustainability, four directions are required to be considered. First, unlike the one-time budget in most of recent literature, a continuous investment/spending model should be formulated and its dynamic balance is valuable to be derived. Second, more attention should be paid to the participator experience. Not only the incentive mechanism but also the cultural recognition can motivate the long-term participant. Third, "Rome was not built in a day." The penetration of MCS will be a gradual process. Current studies always assume that all users or a given probability of users would accept the task allocation, which is not applicable in practice. It is better to define a new feature of penetration to characterize the development of MCS.

Fourth, a green task allocation is valuable in sustainable MCS. Here, the "green" have several meanings including: adopt the green energy, minimize the junk/redundant information, and reduce the human cost. Based on the above directions, we think sustainable MCS task allocation is still an uncovered treasure, worthy of our researching.

### B. Behavioral Models for Improving Task Allocation

Actually, many factors will affect users' behavior in task completion, which is crucial for task allocation. For example, if we can predict the workers' task acceptance likelihood, then we can further optimize the task allocation by assigning more tasks to those more likely to accept it [63]. Literatures of general crowdsourcing predict workers' behaviors by considering factors, such as topical interest, expertise, and time availability. In addition to that, MCS should further consider many other contextual factors. For example, contexts (e.g., the participants' motion and the position of the mobile device) has a significant impact on the sensing data quality for certain types of MCS tasks. We can train a sensing data quality classifier, which extract the relation between context information (such as the participants' motion) and sensing data quality, to estimate data quality in MCS. This classifier can be applied to guide user recruitment and task assignment in MCS. In another example, by detecting instances where a participant is bored, it is then possible to take advantage of their contextual cognitive surplus.

### C. Hybrid MCS Task Allocation

Existing task allocation solutions adopt either the opportunistic mode or the participatory mode (mentioned in Section III). Motivated by the complementary nature of these two modes, there may be a hybrid solution, which can effectively integrate the opportunistic-mode and the participatory-mode task allocation. For example, we can recruit a number of opportunistic workers to complete tasks during their routine trajectories. Then, we further assign some other participatory workers to locations where tasks cannot be completed by the opportunistic workers alone. The hybrid solution has

two advantages. First, from the perspective of the workers, it naturally accommodates the workers' participation preferences and makes full use of the available human sensing resources. Although the workers all want to contribute sensing data to MCS tasks, their 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. Second, from the perspective of the task organizer, it 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. However, when the task allocation of these two types of workers is correlated (e.g., they share a total incentive budget), it is challenging to jointly optimize them, which remains as a future research issue.

### D. Considering Data Sharing Among Multiple Tasks

Existing work for MCS task allocation only considers the competitive relation among multiple tasks. That is, if a sensing resource (workers) is allocated to some tasks, other tasks cannot utilize it. However, we can take into account more complicated situations, where sensing results for a task can be utilized for another task. Intuitively, although the number of sensing tasks may become larger and larger with the popularity of MCS, the kinds of sensors in the smartphone are limited. To this end, some tasks can share the same type of sensing data, or the sensing data among tasks are co-related. For example, the queue time estimation task in [67] needs to use GPS, accelerometers, and microphones, while noise level monitoring task requires GPS and microphones. In this case, the GPS and accelerometers can be shared.

### E. Social-Network-Assisted MCS Task Allocation

Existing studies commonly recruit workers and allocate tasks on specialized MCS systems with assumed large user pools, so that their goal is to select a subset of users from the pool with the consideration of some factors (e.g., sensing quality and cost). However, they fail to work when such assumed large user pools do not exist. In the recent decade, the popularity of mobile social networks (MSNs, e.g., Facebook, Twitter, Foursquare, etc.) has created new mediums for information sharing and propagation, and they have gradually become promising platforms for advertising novel products or innovative ideas. Inspired by the power of MSN, instead of relying on specific MCS platforms, it is interesting to study how to recruit workers of MCS task in a novel manner, i.e., exploiting social network as the task allocation platform. Nevertheless, we cannot directly adopt the information propagation model of the social network in social-network assisted MCS task allocation. When determining whether the user will be influenced by the



propagated information, the existing models merely consider the influence from the neighbors in the social network without taking the specific factors about MCS tasks into account. For example, whether the incentive is attractive or whether the task's topic is interesting would have a significant impact on the users' decision on accepting or declining the task. Thus, it needs to extend the state-of-the-art propagation models in the social network research community by introducing MCS-specific factors.

#### F. Composite MCS Task Allocation

For previous work of MCS task allocation, sensing tasks are rather simple, where a participant's mobile device can provide a complete sample by utilizing a single type of sensor. In the real-world application scenarios, however, there are some other MCS tasks which can be rather complex, which consists of several subtasks and different types of sensors or sensing capability. We refer such complicated tasks as the *composite* MCS tasks. Air quality monitoring task is a typical example of composite MCS because the air quality index (AQI) is calculated based on the sensor readings of multiple types of pollutants, including ground-level ozone, particulates, sulfur dioxide, carbon monoxide, and nitrogen dioxide. A participant usually fails to provide a full sample for a composite task, because he/she may not have the sensing capabilities of all subtasks. For example, their smartphone may not be embedded with the required sensors (e.g., SO<sub>2</sub> sensor), or they deliberately disable the sensors (e.g., microphone) to preserve their privacy. If we assume that the mobile device of each participant is embedded with a subset of the required pollutant sensors, then a complete AQI in a certain place should be obtained through the collaborative sensing among multiple participants. As each participant is only able to complete a subset of subtasks, the composite task should be accomplished through the collaboration of multiple participants. Therefore, the task allocation of the composite task is much more complicated, so that the study on the task allocation of the composite MCS is an important direction for future research.

#### G. Location-Privacy-Concerned MCS Task Allocation

While much theoretical privacy protection has been proposed in MCS task allocation, it seems that in real applications, privacy protection is still often ignored, or implemented by some simple configuration options where users can set private locations to avoid being sensed. This phenomenon may be because users are often hard to understand the real privacy protection effect for them if the privacy mechanism is not intuitively comprehensible. Moreover, in reality, many users may be unclear about the potential consequences incurred by privacy leakage [62], which makes implementing privacy mechanisms is not urgent for MCS business entities. Therefore, there is still a huge gap between the industry and academia in the MCS location privacy concerned task allocation. To fill this gap, one possible direction is to design more user-friendly (understandable) privacy mechanisms and educate the public about the severe privacy leakage consequences, so as to make the users more concerned about their

privacy and get the most appropriate privacy configurations; and another is to make some guidelines and regularizations about user privacy for MCS business entities, so as to facilitate a more secure sensing environment for MCS participants.

#### H. Task Allocation for Sparse MCS

While many MCS task allocation methods are proposed to maximize the sensing coverage of the target area, how to deal with the missing data of un-sensed regions are often neglected in those methods. Recently, researchers have proposed "sparse MCS" paradigm, where the treatment of such missing data in un-sensed regions is formalized as an important stage. State-of-the-art machine learning approaches like matrix completion and compressive sensing are used in this stage to infer the missing data with high quality [39], [40]. In sparse MCS, the target of task allocation differs from coverage maximization, as the sensing data of different regions at different time slots can contribute diversely to the overall missing data inference quality. However, because the ground truth sensing values of un-sensed areas are unknown, how to quantify the data inference quality is really challenging. Rather than directly comparing the inferred data with ground truth, novel methods have to be developed to measure the data inference quality. If more real-life factors are added, e.g., different participants are paid with different incentives, the task allocation for sparse MCS will become even more complicated. To this end, how to design effective and efficient task allocation schemes for sparse MCS needs more research efforts.

#### I. Task Allocation for Indoor MCS

Existing task allocation approaches are mainly designed for outdoor scenarios. MCS in indoor areas is becoming more and more crucial for flow management, security and surveillance, or building usage statistics in recent years. For example, studies such as [74] and [75] proposed floor plan reconstruction and indoor navigation systems by leveraging crowd-sensed data from mobile users. These studies mainly focus on the inference of the floor layout or people's locations given a fixed set of mobile devices and their signals. It is interesting to further study the task allocation problem for these indoor MCS applications. For example, if the candidate users who are willing to share the signals require certain incentive reward, then it is interesting to study how to select a set of devices for jointly optimizing the incentive cost and accuracy of floor reconstruction or indoor navigation.

## VII. CONCLUSION

In this paper, we survey the task allocation problem of a special case of crowdsourcing, named MCS, which requires workers' physical presence at certain locations in order to complete urban environment sensing tasks. We discuss the unique characteristics of MCS. We then classify the state-of-the-art research into different categories with different problem formulation or allocation algorithms. In the end, we suggest several promising issues as future research directions.

## ACKNOWLEDGMENT

The authors really appreciate all the reviewers and editors for their insightful comments, which significantly improve this paper.

## REFERENCES

- [1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.
- [2] A. Antonić, M. Marjanović, K. Pripuzić, and I. P. Žarko, "A mobile crowd sensing ecosystem enabled by CUPUS: Cloud-based publish/subscribe middleware for the Internet of Things," *Future Gener. Comput. Syst.*, vol. 56, pp. 607–622, Mar. 2016.
- [3] J. An, X. Gui, Z. Wang, J. Yang, and X. He, "A crowdsourcing assignment model based on mobile crowd sensing in the Internet of Things," *IEEE Internet Things J.*, vol. 2, no. 5, pp. 358–369, Oct. 2015.
- [4] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011.
- [5] J. A. Burke *et al.*, "Participatory sensing," in *Proc. Int. Workshop World Sensor Web (WSW)*, 2006, pp. 117–134.
- [6] F. Alt, A. S. Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz, "Location-based crowdsourcing: Extending crowdsourcing to the real world," in *Proc. ACM 6th Nordic Conf. Human-Comput. Interact. Extend. Boundaries*, 2010, pp. 13–22.
- [7] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, "TRAC: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *Proc. IEEE INFOCOM*, 2014, pp. 1231–1239.
- [8] J. Rogstadius *et al.*, "CrisisTracker: Crowdsourced social media curation for disaster awareness," *IBM J. Res. Develop.*, vol. 57, no. 5, pp. 1–13, Sep./Oct. 2013.
- [9] M.-R. Ra, B. Liu, T. F. L. Porta, and R. Govindan, "Medusa: A programming framework for crowd-sensing applications," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, 2012, pp. 337–350.
- [10] T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma, "PRISM: Platform for remote sensing using smartphones," in *Proc. 8th Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, 2010, pp. 63–76.
- [11] J. Howe, "The rise of crowdsourcing," *Wired Mag.*, vol. 14, no. 6, pp. 1–4, 2006.
- [12] K. Mao, L. Capra, M. Harman, and Y. Jia, "A survey of the use of crowdsourcing in software engineering," *J. Syst. Softw.*, vol. 126, pp. 57–84, Apr. 2017.
- [13] G. Xintong, W. Hongzhi, Y. Song, and G. Hong, "Brief survey of crowdsourcing for data mining," *Expert Syst. Appl.*, vol. 41, no. 17, pp. 7987–7994, 2014.
- [14] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," *Semantic Web*, vol. 8, no. 3, pp. 489–508, 2017.
- [15] C. Kulkarni, S. P. Dow, and S. R. Klemmer, "Early and repeated exposure to examples improves creative work," in *Design Thinking Research*. Cham, Switzerland: Springer Int., 2014, pp. 49–62.
- [16] E. Mollick, "The dynamics of crowdfunding: An exploratory study," *J. Bus. Venturing*, vol. 29, no. 1, pp. 1–16, 2014.
- [17] A. T. Campbell *et al.*, "The rise of people-centric sensing," *IEEE Internet Comput.*, vol. 12, no. 4, pp. 12–21, Jul./Aug. 2008.
- [18] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Earphone: An end-to-end participatory urban noise mapping system," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2010, pp. 105–116.
- [19] M. Mun *et al.*, "PEIR, the personal environmental impact report, as a platform for participatory sensing systems research," in *Proc. ACM 7th Int. Conf. Mobile Syst. Appl. Services*, Jun. 2009, pp. 55–68.
- [20] B. Hull *et al.*, "CarTel: A distributed mobile sensor computing system," in *Proc. ACM 4th Int. Conf. Embedded Netw. Sensor Syst.*, Oct. 2006, pp. 125–138.
- [21] Y. Chon, N. D. Lane, Y. Kim, F. Zhao, and H. Cha, "Understanding the coverage and scalability of place-centric crowdsensing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Sep. 2013, pp. 3–12.
- [22] K. K. Rachuri, C. Mascolo, M. Musolesi, and P. J. Rentfrow, "SociableSense: Exploring the trade-offs of adaptive sampling and computation offloading for social sensing," in *Proc. MobiCom*, 2011, pp. 73–84.
- [23] S. Reddy *et al.*, "Image browsing, processing, and clustering for participatory sensing: Lessons from a DietSense prototype," in *Proc. ACM 4th Workshop Embedded Netw. Sensors*, Jun. 2007, pp. 13–17.
- [24] D. Zhang, L. Wang, H. Xiong, and B. Guo, "4W1H in mobile crowd sensing," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 42–48, Aug. 2014.
- [25] J. Wang, Y. Wang, S. Helal, and D. Zhang, "A context-driven worker selection framework for crowd-sensing," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 3, 2016, Art. no. 12.
- [26] J. Wang, Y. Wang, D. Zhang, and S. Helal, "Energy saving techniques in mobile crowd sensing: Current state and future opportunities," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 164–169, May 2018.
- [27] N. D. Lane *et al.*, "Piggyback CrowdSensing (PCS): Energy efficient crowdsourcing of mobile sensor data by exploiting smartphone app opportunities," in *Proc. 11th ACM Conf. Embedded Netw. Sensor Syst.*, Nov. 2013, p. 7.
- [28] Y. Wang *et al.*, "A framework of energy efficient mobile sensing for automatic user state recognition," in *Proc. ACM 7th Int. Conf. Mobile Syst. Appl. Services*, Jun. 2009, pp. 179–192.
- [29] B. Guo *et al.*, "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm," *ACM Comput. Surveys*, vol. 48, no. 1, p. 7, 2015.
- [30] M.-C. Yuen, I. King, and K.-S. Leung, "A survey of crowdsourcing systems," in *Proc. IEEE 3rd Int. Conf. Soc. Comput. (SocialCom) Privacy Security Risk Trust (PASSAT)*, Oct. 2011, pp. 766–773.
- [31] B. Guo, Y. Liu, W. Wu, Z. Yu, and Q. Han, "ActiveCrowd: A framework for optimized multitask allocation in mobile crowdsensing systems," *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 3, pp. 393–403, Jun. 2016.
- [32] Y. Liu *et al.*, "TaskMe: Multi-task allocation in mobile crowd sensing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Sep. 2016, pp. 403–414.
- [33] D. Zhang, H. Xiong, L. Wang, and G. Chen, "CrowdRecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Sep. 2014, pp. 703–714.
- [34] H. Xiong *et al.*, "iCrowd: Near-optimal task allocation for piggyback crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 15, no. 8, pp. 2010–2022, Aug. 2016.
- [35] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, and Q. Arshad, "Mobile phone sensing systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 402–427, 1st Quart., 2013.
- [36] C. H. Liu *et al.*, "Energy-aware participant selection for smartphone-enabled mobile crowd sensing," *IEEE Syst. J.*, vol. 11, no. 3, pp. 1435–1446, Sep. 2015.
- [37] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: New insights," *Struct. Multidisciplinary Optim.*, vol. 41, no. 6, pp. 853–862, 2010.
- [38] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2289–2302, Nov. 2013.
- [39] L. Wang *et al.*, "Sparse mobile crowdsensing: Challenges and opportunities," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 161–167, Jul. 2016.
- [40] L. Wang *et al.*, "CCS-TA: Quality-guaranteed online task allocation in compressive crowdsensing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Sep. 2015, pp. 683–694.
- [41] Q. Xu and R. Zheng, "When data acquisition meets data analytics: A distributed active learning framework for optimal budgeted mobile crowdsensing," in *Proc. INFOCOM*, 2017, pp. 1–9.
- [42] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment framework for participatory sensing data collections," in *Proc. Pervasive*, 2010, pp. 138–155.
- [43] A. Singla and A. Krause, "Incentives for privacy tradeoff in community sensing," in *Proc. 1st AAAI Conf. Human Comput. Crowdsourcing*, 2013, pp. 165–173.
- [44] G. Cardone *et al.*, "Fostering participation in smart cities: A geo-social crowdsensing platform," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 112–119, Jun. 2013.
- [45] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *Proc. IEEE INFOCOM*, 2015, pp. 2254–2262.
- [46] S. Hachem, A. Pathak, and V. Issarny, "Probabilistic registration for large-scale mobile participatory sensing," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, vol. 18, 2013, pp. 132–140.
- [47] L. Wang, D. Zhang, Z. Yan, H. Xiong, and B. Xie, "effSense: A novel mobile crowd-sensing framework for energy-efficient and cost-effective data uploading," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 12, pp. 1549–1563, Dec. 2015.
- [48] Z. Song, C. H. Liu, J. Wu, J. Ma, and W. Wang, "QoI-aware multitask-oriented dynamic participant selection with budget constraints," *IEEE Trans. Veh. Technol.*, vol. 63, no. 9, pp. 4618–4632, Nov. 2014.

- [49] J. Wang *et al.*, “Fine-grained multitask allocation for participatory sensing with a shared budget,” *IEEE Internet Things J.*, vol. 3, no. 3, pp. 1395–1405, Dec. 2016.
- [50] J. Wang *et al.*, “PSAllocator: Multi-task allocation for participatory sensing with sensing capability constraints,” in *Proc. ACM Conf. Comput. Supported Cooper. Work Soc. Comput. (CSCW)*, 2017, pp. 1139–1151.
- [51] W. Wang, H. Gao, C. H. Liu, and K. K. Leung, “Credible and energy-aware participant selection with limited task budget for mobile crowd sensing,” *Ad Hoc Netw.*, vol. 43, pp. 56–70, Jun. 2016.
- [52] A. A. Bian *et al.*, “Guarantees for greedy maximization of non-submodular functions with applications,” in *Proc. Int. Conf. Mach. Learning*, 2017, pp. 498–507.
- [53] P. Cheng, X. Lian, L. Chen, J. Han, and J. Zhao, “Task assignment on multi-skill oriented spatial crowdsourcing,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2201–2215, Aug. 2016.
- [54] P. Cheng *et al.*, “Reliable diversity-based spatial crowdsourcing by moving workers,” *Proc. VLDB Endowment*, vol. 8, no. 10, pp. 1022–1033, 2015.
- [55] D. Deng, C. Shahabi, and U. Demiryurek, “Maximizing the number of worker’s self-selected tasks in spatial crowdsourcing,” in *Proc. 21st ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, 2013, pp. 324–333.
- [56] D. Deng, C. Shahabi, and L. Zhu, “Task matching and scheduling for multiple workers in spatial crowdsourcing,” in *Proc. 23rd Int. Conf. Adv. Geograph. Inf. Syst.*, 2015, p. 21.
- [57] M. Ibrahim, M. Chamoun, R. Kilany, M. E. Helou, and N. Rouhana, “Comiqua: Collaborative measurement of Internet quality,” *Ann. Telecommun.*, vol. 73, nos. 3–4, pp. 1–13, 2017.
- [58] V. D. Blondel *et al.*, “Data for development: The D4D challenge on mobile phone data,” *arXiv preprint arXiv:1210.0137*, 2012.
- [59] H. To, M. Asghari, D. Deng, and C. Shahabi, “SCAWG: A toolbox for generating synthetic workload for spatial crowdsourcing,” in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2016, pp. 1–6.
- [60] H. Xiong, D. Zhang, L. Wang, J. P. Gibson, and J. Zhu, “EEMC: Enabling energy-efficient mobile crowdsensing with anonymous participants,” *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, p. 39, 2015.
- [61] H. Xiong, D. Zhang, L. Wang, and H. Chaouchi, “EMC<sup>3</sup>: Energy-efficient data transfer in mobile crowdsensing under full coverage constraint,” *IEEE Trans. Mobile Comput.*, vol. 14, no. 7, pp. 1355–1368, Jul. 2015.
- [62] A. Acquisti, L. Brandimarte, and G. Loewenstein, “Privacy and human behavior in the age of information,” *Science*, vol. 347, no. 6221, pp. 509–514, 2015.
- [63] M. Karaliopoulos, I. Koutsopoulos, and M. Titsias, “First learn then earn: Optimizing mobile crowdsensing campaigns through data-driven user profiling,” in *Proc. 17th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Jul. 2016, pp. 271–280.
- [64] L. Wang *et al.*, “Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation,” in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 627–636.
- [65] L. Pournajaf, D. A. Garcia-Ulloa, L. Xiong, and V. Sunderam, “Participant privacy in mobile crowd sensing task management: A survey of methods and challenges,” *ACM SIGMOD Rec.*, vol. 44, no. 4, pp. 23–34, 2016.
- [66] I. J. Vergara-Laurens, D. Mendez, and M. A. Labrador, “Privacy, quality of information, and energy consumption in participatory sensing systems,” in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, 2014, pp. 199–207.
- [67] J. Wang *et al.*, “Real-time and generic queue time estimation based on mobile crowdsensing,” *Front. Comput. Sci.*, vol. 11, no. 1, pp. 49–60, 2017.
- [68] M. Duckham and L. Kulik, “A formal model of obfuscation and negotiation for location privacy,” in *Pervasive Computing*. Heidelberg, Germany: Springer, 2005, pp. 152–170.
- [69] L. Wang, D. Zhang, D. Yang, B. Y. Lim, and X. Ma, “Differential location privacy for sparse mobile crowdsensing,” in *Proc. IEEE 16th Int. Conf. Data Min. (ICDM)*, Dec. 2016, pp. 1257–1262.
- [70] S. Scellato, A. Noulas, and C. Mascolo, “Exploiting place features in link prediction on location-based social networks,” in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Aug. 2011, pp. 1046–1054.
- [71] P. Cheng, X. Lian, L. Chen, and C. Shahabi, “Prediction-based task assignment in spatial crowdsourcing,” in *Proc. IEEE 33rd Int. Conf. Data Eng. (ICDE)*, Apr. 2017, pp. 997–1008.
- [72] H. To, G. Ghinita, L. Fan, and C. Shahabi, “Differentially private location protection for worker datasets in spatial crowdsourcing,” *IEEE Trans. Mobile Comput.*, vol. 16, no. 4, pp. 934–949, Apr. 2017.
- [73] L. Wang *et al.*, “SPACE-TA: Cost-effective task allocation exploiting intradata and interdata correlations in sparse crowdsensing,” *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 2, p. 20, 2017.
- [74] R. Gao *et al.*, “Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing,” in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2014, pp. 249–260.
- [75] C. Zhang, K. P. Subbu, J. Luo, and J. Wu, “GROPING: Geomagnetism and crowdsensing powered indoor navigation,” *IEEE Trans. Mobile Comput.*, vol. 14, no. 2, pp. 387–400, Feb. 2015.
- [76] K. Kim, H. Zabihi, H. Kim, and U. Lee, “TrailSense: A crowdsensing system for detecting risky mountain trail segments with walking pattern analysis,” *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 3, p. 65, 2017.
- [77] J. Wang *et al.*, “Multi-task allocation in mobile crowd sensing with individual task quality assurance,” *IEEE Trans. Mobile Comput.*, vol. 17, no. 9, pp. 2101–2113, Sep. 2018.



**Jiangtao Wang** received the Ph.D. degree from Peking University, Beijing, China, in 2015.

He is currently an Assistant Professor with the School of Electrical Engineering and Computer Science, Peking University. His current research interests include mobile crowd sensing, collaborative computing, and social computing.



**Leye Wang** received the B.Sc. and M.Sc. degrees in computer science from Peking University, Beijing, China, and the Ph.D. degree from the Institut Mines-Télécom/Télécom SudParis, Évry, France, and the Université Pierre et Marie Curie, Paris, France, in 2016.

He is currently a Post-Doctoral Research Fellow with the Hong Kong University of Science and Technology, Hong Kong. His current research interests include mobile crowdsensing, social networks, and intelligent transportation systems.



**Yasha Wang** received the Ph.D. degree from Northeastern University, Shenyang, China, in 2003.

He is a Professor and the Associate Director of the National Research & Engineering Center of Software Engineering, Peking University, Beijing, China. His current research interests include urban data analytics and ubiquitous computing.



**Daqing Zhang** received the Ph.D. degree from the University of Rome “La Sapienza,” Rome, Italy, in 1996.

He is a Professor with the School of Electrical Engineering and Computer Science, Peking University, Beijing, China, and Télécom SudParis, Évry, France. His current research interests include context-aware computing, urban computing, and mobile computing.



**Linghe Kong** received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2012.

He is currently a Research Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His current research interests include wireless networks, big data, mobile computing, and smart energy systems.