Front. Comput. Sci., 2018, 12(2): 231–244
https://doi.org/10.1007/s11704-017-7024-6

RESEARCH ARTICLE

Mobile crowd sensing task optimal allocation: a mobility pattern matching perspective

Liang WANG (✉)[1,2], Zhiwen YU[1], Bin GUO[1], Fei YI[1], Fei XIONG[3]

1 School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China
2 School of Electrical and Control Engineering, Xi'an University of Science and Technology, Xi'an 710054, China
3 School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China

Abstract With the proliferation of sensor-equipped portable mobile devices, Mobile CrowdSensing (MCS) using smart devices provides unprecedented opportunities for collecting enormous surrounding data. In MCS applications, a crucial issue is how to recruit appropriate participants from a pool of available users to accomplish released tasks, satisfying both resource efficiency and sensing quality. In order to meet these two optimization goals simultaneously, in this paper, we present a novel MCS task allocation framework by aligning existing task sequence with users' moving regularity as much as possible. Based on the process of mobility repetitive pattern discovery, the original task allocation problem is converted into a pattern matching issue, and the involved optimization goals are transformed into pattern matching length and support degree indicators. To determine a trade-off between these two competitive metrics, we propose greedy-based optimal assignment scheme search approaches, namely MLP, MDP, IU1 and IU2 algorithm, with respect to matching length-preferred, support degree-preferred and integrated utility, respectively. Comprehensive experiments on real-world open data set and synthetic data set clearly validate the effectiveness of our proposed framework on MCS task optimal allocation.

Keywords mobile crowd sensing, task allocation, mobility regularity, pattern matching

Received January 19, 2017; accepted August 10, 2017

E-mail: liangwang0123@gmail.com

1 Introduction

With the dramatic proliferation of sensor-equipped portable mobile devices, a novel sensing paradigm named *Mobile Crowdsensing* (MCS) [1,2] has become an effective way to sense and collect data about the environment, human society and individuals. Instead of deploying static and expensive distributed sensors, MCS utilizes built-in sensors in smartphone and human power to acquire environment conditions (i.e., air quality, noise level, emergent events, etc.). Up until now, many MCS-related applications have been emerged, such as travel recommendation [3], public information dissemination [4], queue time estimation [5], online opinion formation [6], urban computing [7–9] and so on.

In MCS applications, there are three players: *requestor* publishes sensing tasks and provides incentive rewards to motivate participants, *participant* is the mobile user who performs sensing tasks and submits measurements to platform, and *platform* is responsible for managing tasks, coordinating requestor and participant, etc. In reality, MCS requestors with limited incentive budget are usually expect to achieve reliable results from collected data set. The role of incentive rewards is to compensate incurred cost of participant users, such as travelling distance, communication overhead, energy consumption and so on. However, this creates a dilemma: when we simply cut down requestors' incentive budget, as the payment is not enough to cover participant users' sensing cost, it will hinder users' participation in MCS, and may bring about risks of participant churn. Consequently, the quality of MCS sensing data could not be guaranteed, and the performance of MCS application will deteriorate observably.

Given an MCS task, the cost incurred in accomplishment phase is diversiform for different candidate participants, due to the various mobility distribution over spatiotemporal dimensions, devices status and so on. Recent studies have established that spatial location holds a dominant position in participants' potential cost, and also plays an important role in the quality of MCS results [10,11]. Therefore, how to decrease sensing cost of MCS task, particularly the travelling cost, has become a critical issue to improve the whole resource efficiency, both lighten the burden of participant users and requestors. Especially, under multi-task allocation circumstances [12–15], this problem will become more complicated and challenging considering the spatiotemporal diversity and involved constraints.

We make an important observation that it can significantly decrease travelling cost by taking advantage of a *synchronous sensing manner*, which is to allocate MCS tasks according to participant users' potential moving routes. As illustrated by the example in Fig. 1, there are three participant users u_1, u_2 and u_3, and five sensing tasks, t_i, $1 \leqslant i \leqslant 5$, which are distributed over different spatial locations. Assuming that participant u_1 will pass loc_5 and loc_6 in the near future, hence it is appropriate to allocate t_4 and t_5 to him. However, if the sensing process is conducted specially, for example, allocating task t_1 to user u_1, u_1 will have to change his scheduled route. It is clear that extra traveling distance denoted by red dotted line will be imposed to u_1. Accordingly, u_1's sensing cost will increase, and requester's budget cost will also grow relatively. Actually, allocating t_1 to u_1 is essentially a local optimal approach, which is nearest first strategy used commonly in the existing studies [12–14,16].

The previous observation motivates us to align MCS task with participant users' moving route, and so the sensing process can be regarded as a "by-product" compared with users' main business. In this way, extra movement distance participant users travel around to perform tasks can be avoided, and requestor's budget cost will also be reduced significantly. More importantly, as the assigned task is conducted during users' scheduled route, the quality of collected sensor readings could be improved, due to the phenomena that users fake measurement readings without visiting specific locations will be restrained. In addition, we hire a bundle strategy to further improve the resource efficiency, which is to maximize the number of assigned tasks per user. In this case, sensing activity cost per task can be reduced, which is consistent with scale effect in economics and will be verified by a questionnaire investigation in later part. Therefore, our focus shifts to assign tasks to participant users as much as possible based on participant users' moving route.

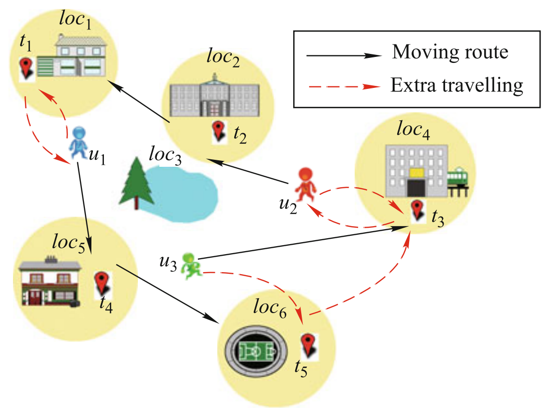


Fig. 1 A scenario of MCS tasks allocation

In response to the above-mentioned concerns, we explore the relationship between MCS task allocation and user mobility regularity in this papaer, and introduce a resource-efficient MCS task allocation framework. More specially, by discovering the implicit mobility regularity from historical traces, the original task allocation problem is transformed into the form of matching discovered mobility patterns with MCS task sequences, with the goal of minimizing sensing cost and maximizing sensing quality. Based on this framework, we propose efficient allocation algorithms taking into account of cost budget and sensing quality. The main technical contributions made in this paper are summarized as follows:

- To accommodate MCS task optimal allocation, we formulate the implicit mobility regularity as user-support mobility repetitive behavior, and leverage a breadth-first strategy to discover the behavior patterns.
- By aligning task sequences with user mobility route, we convert and formulate the task allocation problem into pattern matching issue. Under pattern matching scenario, with respect to two competing optimal indices, we devise different greedy-based task sequence matching algorithms with the objectives of achieving resource efficiency and quality assurance.
- We conduct extensive experiments using real-world and synthetic data sets to evaluate our proposed algorithms. The results show the effectiveness and efficiency of our proposed task allocation problem solving approaches.

The rest of this paper is organized as follows. Section 2 discusses related work and verifies our hypothesis via a questionnaire survey. Section 3 characterizes crowd participant users' implicit mobility repetitive behavior regularity. Section 4 gives the problem definition, transforms original problem into pattern matching issue, and proposes effective allocation algorithms to optimally distribute MCS tasks. We report our experimental results in Section 5, and conclude our study in Section 6.

2 Related work and hypothesis validation

2.1 Related work

In MCS applications, the platform recruits participant users to complete released tasks. Considering the requirements of sensing task (e.g., temporal and spatial constraints, background knowledge, etc.), not everyone is well qualified to undertake a specific task. Roughly speaking, how to select an appropriate user from participant candidate pool to perform a given sensing task is a critical problem, it can directly affect the final sensing quality. There has been recently works on MCS task allocation and participant selection. Among them, two types of task allocation have been studied in previous works.

• Single task allocation Studies in this stream allocate a single task to candidate participant users [17–20]. Reddy et al. [17] propose a recruitment framework to identify well-suited participant users for data collection according to spatial-temporal availability and participation habits. Pournajaf et al. [18] propose a dynamic data driven spatial crowd sensing task assignment model by building a synergistic feedback loop between application simulations and data collection. Zhang et al. [19] devise a participant selection framework to assist task requestors to identify users, with an optimization objective to minimize the budget payments. In [20], Xiong et al. propose a task allocation framework, named iCrowd, to achieve dual optimal goals which are k-depth coverage maximization and incentive payment minimality.

• Multi task allocation Studies in this stream assign multi tasks to candidate participant users [12–15,21,22]. Kazemi et al. [12] define a maximum task assignment problem, in which participant users report their current location to platform and platform allocates to each user his nearest tasks with the goal of maximizing the overall number of assigned tasks. He et al. [13] study the problem of location-dependent task allocation with time budgets for participant users. Based on bargaining theory, they devise an approximation algorithm to maximize the rewards of platform. Liu et al. [14] study the multi-task allocation problem and propose a participant selection framework, TaskMe. In [14], two typical situations are investigated, that is few participants, more tasks and more participants, few tasks. Based on the Minimum Cost Maximum Flow theory, two optimal algorithms are proposed to achieve the optimization goal. Kandappu et al. [22] conduct experiments in campus environment to investigate the user skew and results veracity problem.

The most relevant work to this paper includes [15,21]. In [15], a statistical result of each participant user's historical locations is used to derive the probability that user will pass by a specific location in the next day. It also follows mobility prediction in task allocation process, but only considering isolated location prediction instead of moving sequence (i.e., moving route). Moreover, it strives to optimize the cost budget and fixes data quality as a pre-specified upper bound. While our work attempts to optimize these two involved objectives simultaneously. By leveraging a hypothetical model TLW, Hachem et al. [21] predict mobile users' future locations in next time time window. From the prediction they choose a minimal number of participants expecting to achieve a certain coverage in target area in the next time slot. However, the mobility estimation model assumes that users have a constant speed and a uniformly distributed direction. Obviously, it may not correspond to reality, and is inadequate to characterize complicated practical scenarios. Therefore, the prior solutions could not be directly applied into our work.

2.2 Hypothesis validation

In order to verify our proposed hypothesis that synchronous sensing manner can achieve resource efficiency, we conduct a questionnaire survey. We define two different sensing manner: special manner and synchronous manner, in which the former is to complete tasks specially, and the latter one is to complete tasks during their moving routes. Some virtual crowd sensing tasks (such as noise measurement, taking photo) are provided, and subjects are required to answer questions or make a choice among some given options. The results show that about 73.67% believe there exist obvious differences between these two sensing manners.

• Sensing task price Given a sensing task, subjects are required to declare a price among four given options, such as 1–5, 6–10, 11–15 and 16–20 dollars. The resultant distribution of declared prices is reported in Table 1. Plainly, it shows that the average price in special sensing manner is 1.885 times more than synchronous manner's. Among 73.67% survey respondents whose opinion supports our hypothesis, the differences between these two sensing manner rise up to 2.76 times.

Table 1 Sensing cost investigation

	1–5 dollars	6–10 dollars	11–15 dollars	16–20 dollars
Special manner	12	28	11	10
Synchronous manner	41	9	1	6

• Multiple task price To investigate the multi-task situation, we devise a mobility route, such as {*library* → *restaurant* → *public garden* → *gymnasium*}, and four sensing tasks are associated with these involved locations. Assuming that respondents will follow the given moving routine in the near future, at each time we allocate each participant one, two and three tasks, respectively. The subjects are required to declare a price at every turn. Finally, we found that 74.2% subjects' declared prices of multi-tasks are less than the cumulation of relative micro-tasks' price. More specifically, the results are reported in Table 2, where $\overline{Price}$ denotes average price of the 74.2% subjects. The average prices of task 1, task 2 and task 3 are \$ 11.10, \$10.70 and \$ 10.34, respectively. The multi-task's price of task 1 and 2 is \$16.625, which is less than the summation of these two tasks, such as $11.10 + 10.70 = 21.80$, by about 28.51%; and the average price of whole tasks is less than the cumulation of these three tasks by about 32.57%. In other words, the more allocated tasks to a participant user, the lower the average price per task. This phenomenon can be explained by the theory of Scale Economies Effect, and some recent research also support it [22,23]. Therefore, in order to decrease sensing cost, an effective strategy is to assign more tasks per user. That is to say, given sensing task set, if the number of selected participant users can be reduced, we can achieve the optimal goal of minimizing the total sensing costs.

Table 2 Task price in multi-task allocation situation

	Task 1	Task 2	Task 3	Task 1+2	Total tasks
$\overline{Price}$/\$	11.10	10.70	10.34	16.625	21.67

3 Characterizing mobility repetitive regularity of participant users

In this section, we exploit and discover the implicit moving regularity of participant users from historical behavior recordings. Then according to the obtained moving regularity, we can achieve a predicted mobility route of volunteer participants' in the future.

3.1 Mobility repetitive behavior modeling

In reality, human user's mobility behavior usually exhibits a repetitive characteristic to some extent. For example, a moving behavior, such as: *leave home* → *arrive at office* → *shopping at supermarket*, may repeats periodically from day to day. Obviously, by leveraging human user's mobility observations, it is possible to statistically characterize and extract this repetitive mobility regularity for each user [24,25]. Let $TD = \{(u_k, loc_1, time_1), (u_k, loc_2, time_2), \ldots, (u_k, loc_n, time_n)\}$ be the original movement database of a mobile user u_k, in which the samplings are recorded in chronological order, and an element $(loc_i, time_i)$, $i = 1, 2, \ldots, n$, implies that u_k is located at position loc_i at time instance $time_i$. Given a set of trace recordings, in this paper, our primary job is to model and discover participant's repetitive mobility regularity in order to accommodate MCS task allocation problem.

Before formulating the mobility regularity, we firstly define some conceptions to elaborate the hidden mobility regularity. An *observation period* P is defined as a time duration across users' historical observed movement traces. Given a specific observation period P, for each participant user, it will be split into many non-overlapped *cycles* with the equal length of C_{len}. In addition to accommodating MCS application setting requirements, divided cycle C can represents how long a repetitive mobility behavior periodically repeats once again. For example, given a historical trace data set across one week, if the length of divided cycles C_{len} is specified as "one day", we can obtain seven time slots accordingly. However, in practice, each participant user could not appear in every cycle, and participant users' trace records are inhomogeneous over split sensing cycles (i.e., some active users frequently appear in trace history; while some inactive users appear occasionally in records) [26,27]. To represent this characteristic, in this paper, split cycle in which u_k does indeed appear is named as *valid cycle*; while the cycle does not record u_k's mobility trace is *invalid cycle*. As we only need to focus on valid cycles for each participant user, given a specific observation period, each user's original trace data set will be divided into N valid cycle subsets in which the time interval equals to C_{len}. In order to depict the frequency of mobility behavior in trace history, by borrowing the conception in frequent itemset mining, we employ *support* to represent each mobility behavior's recurrence rate throughout valid cycles. Formally, it is defined as the fraction of N cycles that contain this specific behavior to the number of divided cycles C.

Obviously, due to the properties described above, the repetitive mobility regularity or behavior is different from common period pattern mining process. 1) The period pattern mining only focuses on the knowledge of frequent pattern itself, but ignores the pattern's contributor. However, in our scenarios, as each MCS task must be allocated to a specific participant user, the information of behavior pattern's holder must be recorded. 2) As different mobile users may support an identical mobility pattern, but the corresponding support values for each user may be quite diverse. Thus, the corresponding support value of one specific frequent pattern should also be noted. However, the period pattern mining does not distinguish the frequent patterns whose support is no more than a given threshold value. 3) As explained above, participant users may have different valid split cycles in trace records, on the one hand, the number of valid cycles reveals the active degree of participant users; on the other hand, to some extent, it also indicates the probability of one participant will appear in the future sensing cycle. Therefore, the number of valid cycles must also be associated with any of repetitive mobility behavior. That is to say, in our approach, repetitive mobility behavior, support degree, corresponding contributor and valid cycle all must be correlated to each other. Hence, an appropriate index structure should be devised to organize mobility pattern.

Now, we define the above-mentioned repetitive mobility regularity as the *user-support mobility repetitive behavior* as follows.

Definition 1 A behavior mp which is represented as the form of $\{loc_i \rightarrow loc_j, sup, u_k, N(u_k)\}$ is a repetitive mobility pattern, if it occurs no more than the user-specified threshold $minsup$ times across $N(u_k)$ split cycles in user u_k's original training data set. In mp, the symbols loc_i and loc_j represent different spatial locations, and the symbol sup denotes its support degree which can be formulated as follows:

$$sup = \frac{fre(mp)}{N(u_k))}, \tag{1}$$

where $fre(mp)$ denotes the frequency of occurrence of mobility behavior mp. Note that the repetitive mobility pattern $\{loc_i \rightarrow loc_j, sup, u_k, N(u_k)\}$ denotes that mobile user u_k moves from location loc_i to loc_j, and this behavior repeats sup times in u_k's $N(u_k)$ valid sensing cycles.

3.2 User-support mobility repetitive behaviors discovery

In the following, we will elaborate the discovery process of user-support mobility repetitive behaviors. First of all, a sensing cycle should be determined to accommodate the requirements of MCS tasks. As in this paper, we focus on no-time critical sensing task, i.e., the task can be completed without emergency. Thus, we specify the sensing cycle with the length of 24 hours (We suppose that every sensing task can be completed with one day in this paper). If it is applied into different situation, it can be tuned accordingly.

For each participant user, we construct a mobility behavior subspace Ω which contains his/her mobility regularity. In the process of repetitive behavior discovery, the search operation is conducted in every subspace Ω with such a parallel manner. Taking advantage of breadth-first search strategy [28], we employ the pattern growth approach which includes *candidate pattern generation* and *data set scanning* to extract mobility repetitive patterns one by one. More specifically, we make multiple scans over the N partitioned subsets (i.e., split cycles) to recognize hot locations which are visited frequently by the current user. And then, 1-length repetitive pattern, such as $\{loc_i, sup, u_k, N(u_k)\}$ is recorded. Afterwards, the discovered mobility repetitive patterns found in the previous pass are used to generate candidate repetitive patterns. And by comparing with a given minimum support threshold $minsup$, we can determine which candidate mobility patterns are actually repetitive patterns. This discovery process will terminate when there are no mobility repetitive patterns at the end of one traversal, or no candidate patterns can be generated. The discovery of mobility repetitive behavior is formulated as shown in Algorithm 1, where the symbols Q and L denote the candidate repetitive patterns and real patterns, respectively.

Algorithm 1 Mobility repetitive pattern discovery

Input: Crowd trace data set TD, split cycle C, support threshold $minsup$.
Output: Mobility repetitive patterns $Mp = \{mp_i, i = 1, 2, \ldots, n\}$.
for *each mobile user* u_k **do**
  Construct mobility behavior subspace $\Omega(u_k)$;
  Recognize hot region set L_1;
  $L_1 = \{q \in Q_1 | c.sup \geqslant minsup\}$;
  repeat
    Q_i = Generate candidate from L_{i-1};
    Calculate Q_i's support sup;
    $L_i = \{q \in Q_i | q.sup \geqslant minsup\}$;
    $i = i + 1$;
  until $Q_i \neq \varnothing$ or $L_i \neq \varnothing$
  $Mp = Mp \cup Mp(u_k)$;
end

4 MCS task allocation based on pattern matching

4.1 Problem definition

The mobile participants are denoted by a set of $U = \{u_1, u_2, \ldots, u_n\}$, and sensing tasks to be performed are represented by $T = \{t_1, t_2, \dots, t_m\}$, in which $t_i = (loc_{t_i}, time_{t_i})$ represents that task t_i's spatial location is loc_{t_i}, and published time stamp is $time_{t_i}$. Note that the platform assigns existing tasks according to their chronological orders (i.e., $time_{t_i}$) with batch form.

• **Problem** According to users' mobility repetitive regularities, MCS platform strives to assign published tasks to suitable participants, with the goal of minimizing overall sensing cost, while maximizing the quality of returned readings. For mobile user u_k, suppose sensing tasks which are allocated to him is represented as $TU_k = \{t_1^k, t_2^k, \dots, t_l^k\}$, thus the rewards paid to user u_k is $Cos(TU_k)$. With respect to the quality of MCS data, it is defined as task coverage degree which is the probability that user u_k will visit the physical location loc_{t_i}. For u_k, the coverage degree of assigned tasks TU_k is formulated as: $Cov(TU_k)$, which equals to support value *sup* of mobility pattern $t_1^k \rightarrow t_2^k \rightarrow \cdots \rightarrow t_l^k$.

• **Constraints** 1) Although our goal is to assign more tasks to each user, considering accomplishment capability and resulting *monopoly* among all users, we restrict a maximum workload constraint which denotes the maximum number of allocated tasks per user, i.e., upper bound g_k. For simplicity, in this paper, each user's upper bound value equals to a constant value g.

2) Moreover, in order to guarantee the *reliability* of collected sensing data for each task t_i, a popular method is to gather more than one readings from each MCS task, so a more truthful value can be achieved by aggregating the collection of returned samplings [16,19,20]. In our scenario, we assume that each task t_i needs to collect c_i independent measurements to ensure the sensing quality, where the parameter c_i is determined by the specific requirements of task t_i. For complicated sensing tasks which cover more area, the corresponding independent measurements c_i should be larger to comprehensively understand the target area. In practice, independent measurements of a MCS task can be regarded as a set of different sensing tasks which are associated with same spatial location. It is note that *one volunteer participant could not be allocated a same sensing task more than once*.

Then, our problem can be formulated as following:

$$\begin{cases} \min : \sum_{k=1}^{n} Cos(TU_k) = \sum_{k=1}^{n} Cos(\sum_{i=1}^{l} t_i^k), \\ \max : \sum_{k=1}^{n} Cov(TU_k) = \sum_{k=1}^{n} Cov(\sum_{i=1}^{l} t_i^k), \end{cases} \tag{2}$$

$$\text{s.t.} \begin{cases} \sum_{i=1}^{l} t_i^k \leqslant g, (1 \leqslant k \leqslant n), \\ \sum_{k=1}^{n} t_i^k \equiv c_i, (1 \leqslant i \leqslant m). \end{cases} \tag{3}$$

Obviously, there are dual-objective to be optimized in our problem, which are overall cost budget $\sum_{k=1}^{n} Cos(TU_k)$ and coverage degree $\sum_{k=1}^{n} Cov(TU_k)$. In most cases, these two objectives are competing. For constrain conditions, $\sum_{i=1}^{l} t_i^k \leqslant g$ denotes maximum workload g that user u_k can be assigned maximum tasks each time, while the second equation $\sum_{j=1}^{n} t_i^j \equiv c_i$ represents that the independent measurements of a sensing task t_i should be equal to the number of duplicates c_i.

4.2 Problem convertion

As explained above, via matching the allocated sensing tasks with user's mobility repetitive patterns, the resource efficiency of MCS platform can be improved. Concretely, with respect to the optimization dual-objective, we will transform the original optimization problem under the framework of mobility behavior pattern matching. Firstly, let's consider the cost budget minimum goal. Obviously, if more tasks are matched with user u_k's predicted moving route, the incurred sensing cost would be reduced. In other words, it is equivalent to achieving a longer task sequence matching with mobility pattern. However, in essence, the *sup* value of longer mobility patterns is usually lower than the short ones, because longer pattern usually has low repetition rate compared with short pattern. This means that if we just pursuit long pattern matching (i.e., minimizing cost budget), the support degree of obtained task sequence assignment may decrease. As a result, the second optimization goal, task coverage degree, will reduce accordingly.

In the following, we will employ a toy example to demonstrate it. Given a set of MCS tasks which are located at $loc_1, loc_2, loc_3, loc_4$, from the perspective of minimizing cost budget goal, it is sensible to choose mobility pattern mp_1 as it can match longer task sequence (i.e., all existing tasks). However, its corresponding coverage degree (i.e., 0.55) is the lowest among these three available patterns. If pattern mp_2 is chosen, we can obtain the maximum coverage degree, but the incentive cost is not optimized compare with mp_1. Thus, it is necessary to find a trade-off between these two competing metrics.

Under the pattern matching paradigm, the original optimization problem in Eq. (2) would be converted into another form as follows.

$$\begin{cases} \max : \sum_{k=1}^{n} matlg(TU_k, Mp(u_k)), \\ \max : \sum_{k=1}^{n} matdg(TU_k, Mp(u_k)), \end{cases} \tag{4}$$

where $matlg(TU_k, Mp(u_k))$ and $matdg(TU_k, Mp(u_k))$ denote the matching length and matching support degree between user u_k's allocated sensing task bundle TU_k and his mobility repetitive pattern set $Mp(u_k)$.

4.3 Optimal task allocation approaches

4.3.1 Allocation solution structure

For a given MCS sensing task set $T = \{t_1, t_2, \ldots, t_m\}$, we employ a matrix structure ATU, as shown in Eq. (5), to record and update the task allocation solutions. In ATU, the number of rows denotes candidate participant user pool scale, while the column is the size of tasks to be allocated. If $z_{i,j}$ equals to 1, it means that task t_j is allocated to user u_i; otherwise, it is not. The sum of elements in each row i represents the number of tasks which are allocated to user u_i (i.e., $\sum_{j=1}^{m} z_{i,j} = |TU_k|$), and the sum of each column equals to the value of independent measurements of task t_j (i.e., $\sum_{i=1}^{n} z_{i,j} = c_j$). By this calculation, it is easily to verify whether the constraints of the optimization objective in Eq. (3) are satisfied.

$$ATU = \begin{bmatrix} z_{1,1} & \cdots & z_{1,m} \\ \vdots & & \vdots \\ z_{n,1} & \cdots & z_{n,m} \end{bmatrix}. \tag{5}$$

4.3.2 Task allocation workfolw

Towards these optimization objectives described in Eq. (4), we can prove that the task allocation problem is a NP-hardness problem via a reduction from the classical Travelling Salesman Problem (due to space limit, we do not present here). Hence, in this paper, we devise optimal algorithm which is greedy-based to iteratively derive the desired solutions.

More specially, in each allocation iteration, we take a subset of sensing tasks T^*, in which there is no reduplicate elements (i.e., each task has only one independent measurement in T^*), to select specific participant users. Then the existing mobility repetitive patterns discovered in advance will be retrieved to search for the matched patterns with task sequences in T^*, ordered by tasks' released time stamp. This matching process is substantially a comparison calculation between matching length and matching support degree. If tasks in T^* are matched entirely or partially by a repetitive pattern mp, we say that task set T^* can be matched with this pattern. The number of elements which are shared with T^* and mp is regarded as the matching length $matlg$. For the given task set T^*, there may be more than one mobility patterns can be matched with it. Among these matched mobility patterns, their matching length and corresponding support are diverse as shown in Fig. 2. To find a trade-off between these two metrics, we must synthetically consider matching length and support degree. The corresponding search strategies will be described in detail in the next section.

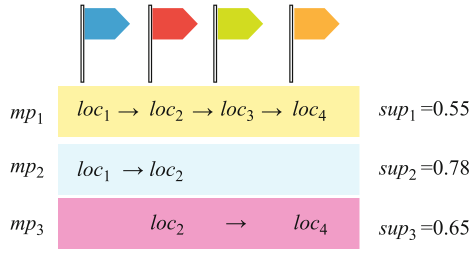


Fig. 2 Mobility patterns matching with different support values

According to the evaluation calculations for each matched patterns, one with maximum evaluation will be selected as the winner, and the corresponding elements shared with T^* will be allocated to the holder (i.e., candidate participant user) of chosen pattern. In T^*, tasks which have not been allocated in this iteration will be returned to the original task set T. Also, the allocated tasks' independent measurement will be reduced by one. If the number of independent measurements for task t_j becomes zero, we exclude t_j from T. For participant user u_i, we assign value one to entry $z_{i,j}$ in matrix ATU if task t_j is allocated to u_i. If u_i's allocated tasks reaches the upper bound g, his mobility patterns will be eliminated from the candidate pattern sets. This allocation process is repeated until no task t_j exists in T. The workfolow of task allocation procedure is shown in Fig. 3.

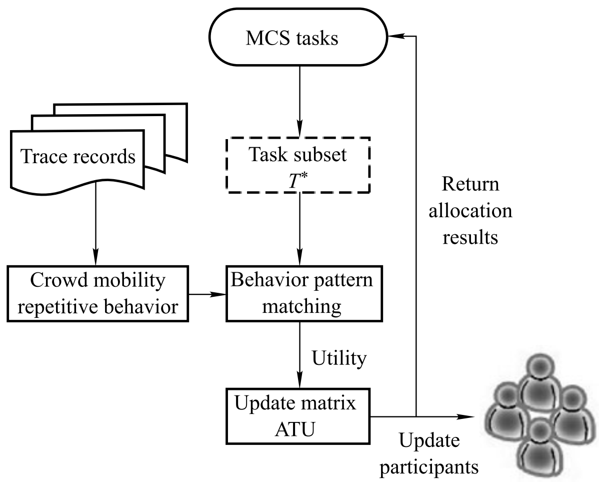


Fig. 3 Workflow of optimization task allocation

4.3.3 Task allocation strategies

Based on the indicators of matching length $matlg$ and support value $matdg$, how to compare different matched patterns' fitness with respect to task set T^* becomes a key issue in our task allocation framework. In this section, with respect to these two metrics, we will devise different evaluation functions to guide the task allocation process. In the following, four different utility calculation strategics are proposed.

1) **Matching Length-Preferred (MLP)** To be specific, if the matching length *matlg* is preferred (i.e., *matlg*-first strategy) , the matching degree of patterns with larger *matlg* values will be greater. In other words, this strategy only concerns with minimizing sensing cost budget objectives, without considering the task coverage degree index.

2) **Matching Degree-Preferred (MDP)** This strategy is the complete opposite of the first MLP strategy. This strategy's focus is on coverage degree index, so one matched pattern *mp* with more matched support value *matdg* will be given greater utility value.

3) **Integrated Utility-1 (IU1)** The IU1 strategy considers both two optimization objectives simultaneously, in which we strive to optimize sensing cost goal with coverage degree as a pre-specified threshold ∂. For any two matched patterns, one pattern whose *matdg* value is less than threshold ∂ will be eliminated firstly. If these candidate patterns' support value are both larger than ∂, the one having longer matching length *matlg* will be chosen as the final assignment.

4) **Integrated Utility-2 (IU2)** Similar to the third strategy, IU2 also incorporates both these two optimization objectives. It employs a product form as $matlg * matdg$, candidate patterns whose value of $matlg * matdg$ is maximum will be selected as the resultant assignment.

According to different utility construction strategies, we devise four task allocation algorithms, namely MLP, MDP, IU1 and IU2. Obviously, these four proposed algorithms' basic assignment procedures are similar, as shown in Fig. 3, the only different part is how to build utility function. Thus, we will not present all algorithms' pseudo code but only MLP in Algorithm 2.

5 Performance evaluation and discussion

In this section, We systematically evaluate the performance of our proposed techniques using a real-world data set and a synthesized data set. Our experiments and latency observations are conducted on a standard server (Windows), with Intel Core i3-3110M CPU, 2.40 GHz and 4 GB main memory.

5.1 The data set

An open WTD data set which is published by researchers at UCSD [26] is utilized as the experimental data set. In campus environment, WTD records all access points (AP) sensed by the user every 20 seconds. Due to the sparse distribution in WTD data set, we filter some APs and users having too little recordings, and obtain a final data set which contain 124 APs and 68 mobile users. In the stage of mobility model training, we employ a 30-day period sampling data set to extract mobility repetitive behavior regularity. Based on the behavior regularity, a set of MCS tasks will be assigned to available users. As each AP has a unique identification number, so we can extract mobile users' behavior patterns related to APs. Moreover, the location of participants and sensing tasks are both related to AP's id.

Algorithm 2 MLP algorithm

```
Input: A set of MCS tasks T, upper bound g, mobility repetitive pattern set Mp.
Output: A task allocation matrix ATU.
Initialize matrix ATU;
repeat
  T* = non-repetition(T);
  MatP = pattern-matching(T*,MP);
  for each matched pattern in MatP do
    Calculate matching length with matlg;
  end
  Select the target pattern with maximum matlg;
  if selected user's allocated task no more than g;
  then
    Allocate MCS tasks based on target pattern;
    Update ATU and T;
  end
until T ≠ ∅;
```

Considering the small-scale and sparse distribution of WTD data set, in order to comprehensively verify our proposed approaches on a larger scale, we synthesize a set of simulation data set based on the real data. More specially, following the statistical probability of users' behavior in WTD, additional 100 virtual mobile users are produced (i.e., 168 available users), and the relevant APs have also increased up to 284.

5.2 Experiment setting and baselines

5.2.1 Experiment setting

In sensing task allocation phase, we employ the mobility periodic pattern base obtained under the condition of $minsup$ = 0.5. The value of pre-specified threshold ∂ in *IU1* equals to 0.7. In these experiments, we investigate the performance of our proposed assignment approaches by varying the scale of sensing tasks, maximum workload parameter g. We use the following means of measurement to evaluate the performance of our proposed optimal task allocation approach, including *average allocated tasks per user*, *average coverage degree per task*, *task allocation efficiency* and *task completion ratio*. Among these above-mentioned four measurements, average allocated tasks per user is used to evaluate cost budget indirectly. Given a set of MCS tasks, the more average allocated tasks per user, the less overall cost budget. The task allocation efficiency is hired to measure the operational efficiency of proposed algorithms. The task completion ratio is to evaluate the completion results of different allocation schemes.

5.2.2 Baseline algorithms

Most existing MCS task allocation work leverages a statistical model to characterize users' mobility regularity and predict their future location [15,20,31]. More specifically, a statistical probability is utilized to predict whether mobile user u_k will pass a given isolated location loc_j. Based on Bayesia' rule, statistical probability $p_{k,j}$ can be calculated as follows:

$$p(loc_j|u_k) = \frac{p(loc_j, u_k)}{p(u_k)}. \tag{6}$$

According to the location probability estimation $p(loc_j|u_k)$, it is reasonable to choose participant user u_k with probability $p(loc_j|\cdot)$ which is more than a specified threshold P_{thld} for task location loc_j. Based on the statistical probability-based MCS task allocation approaches [29–31], we adapt two baseline algorithms, including increment greedy-based assignment (i.e., *IGA*) and bipartite graph-based assignment (i.e., *BGA*), to accommodate our problem scenario and evaluate the performance of proposed approaches. And the threshold value P_{thld} in statistical probability model is set to 0.5.

5.3 Experiment and result analysis

5.3.1 Frequent mobility periodic pattern discovery

Based on WTD data set, the mobility periodic pattern discovery technique is evaluated by varying the support threshold value (60%–90% with 10% increment). The corresponding results are reported in Fig. 4. The number of discovered mobility patterns versus the support values from 60% to 90% are shown in it, in which the vertical axis is represented with logarithmic scale, such as 10^1, 10^2, 10^3 and 10^4. Obviously, it is observed that the number of discovered periodic patterns increases exponentially as the support *minsup* reduces. The reason is plain that when the support value increases, less mobility patterns will satisfy the given requirements. In order to remove the redundant patterns and improve the matched pattern retrieval efficiency, we pick out the closed patterns of each user (i.e., *user intra closed patterns*) and all participant users (i.e., *user inter closed patterns*). For each given support value, the number of discovered total mobility periodic patterns is largest, followed by *user intra closed patterns* and *user inter closed patterns*.

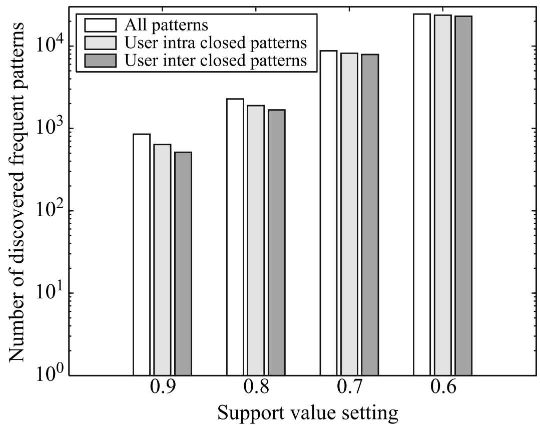


Fig. 4 Discovered frequent patterns

5.3.2 Varying the number of sensing tasks

In this part, we investigate the performance of proposed assignment approaches by varying the number of sensing tasks in real-world and synthetic data set. Figures 5 and 6 show the results under several different sensing task sizes. Here, "20–38" means that the number of tasks is 20 and the corresponding independent measurements are 38. The maximal independent measurement size is set to 3, and the maximum workload per user equals to 6. For each situation, we conducted ten times of experiments to get the average values. In Figs. 5 and 6, we report the performance comparison on *average assigned tasks per user* and *average coverage degree per task* for our proposed approaches and two baseline algorithms.

As shown in Fig. 5, MLP algorithm outperforms the rest of task allocation approaches, followed by IU2, BGA, IU1, MDP and IGA approach. The reason is that MLP algorithm employs a longest matching pattern-first strategy to assign tasks, it can assign more tasks to each participant user at one time according to the discovered repetitive pattern's length. In our cases, the maximum assigned tasks per user can achieve the maximum workload setting. Thus it can achieve best performance with respect to the metric of average assigned tasks per user (i.e., cost budget objective). MDP algorithm achieves poor performance with this metric due to the fact that it adopts matching-degree preferred strategy, so it usually does not prefer matching length but support degree. While the integrated utility-based approaches, IU1 and IU2 can be regarded as a compromise between MLP and MDP algorithms.

The IGA approach does not utilize moving route regularity, but only a location-user pair (loc_{t_i}, u_j) to predict the potential probability. By employing an increment assignment strategy, it only determines one user-task pair in each iteration, so the average assigned task is less than other algorithms. The BGA algorithm searches connection edges between user node and task node in graph construction. By hiring a strategy of user with maximum connections first, it achieves better performance compared with most approaches.

per user (i.e., the cost budget). This is coincidence with the above discussion that these two measurements are competing. Among our proposed four strategies, MLP and MDP algorithms are two extremes with respect to cost budget and coverage degree. While IU1 and IU2 obtain a trade-off between these two measurements. Moreover, as IGA algorithm ranks the qualified users for one specific task according to the predicted visiting probability (i.e., coverage degree), it also achieves a better overall performance of coverage degree.

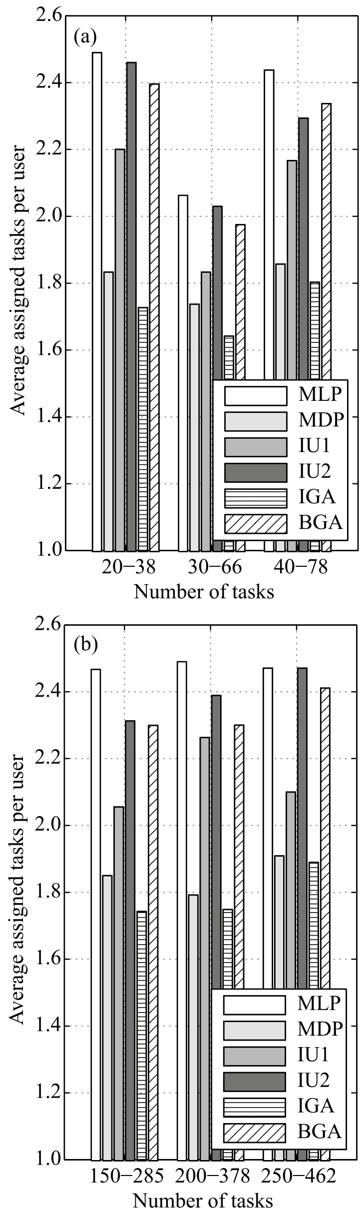


Fig. 5 Average assigned tasks per user. (a) Real-world data set; (b) synthesized data set

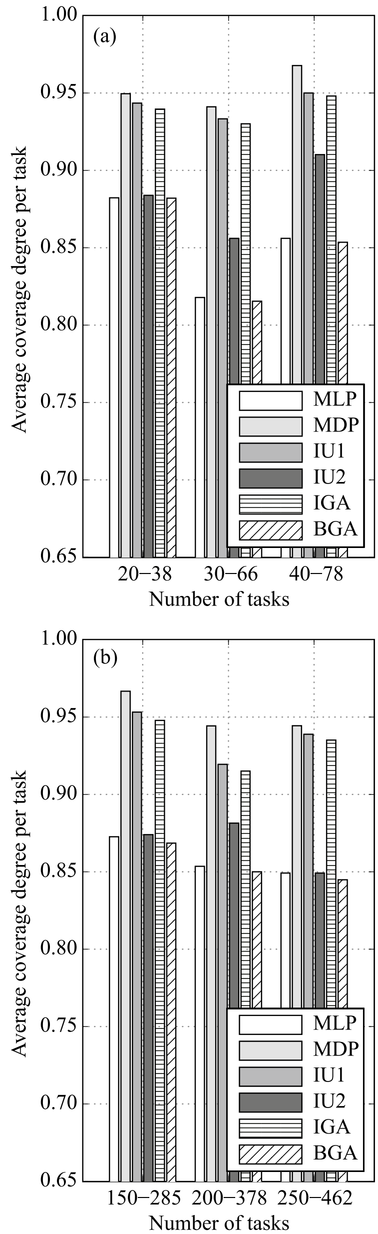
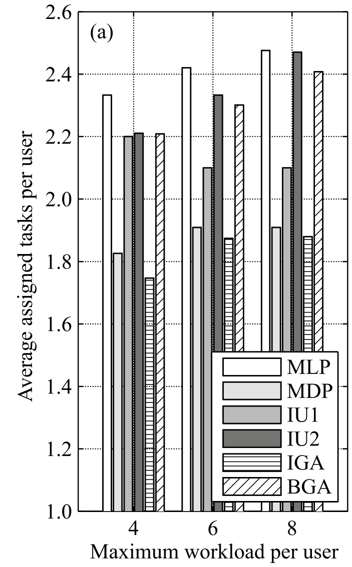


Fig. 6 Average coverage degree per task. (a) Real-world data set; (b) synthesized data set

With regard to the measurement of *coverage degree*, the corresponding results are listed in Fig. 6. It is interesting to note that the results are opposite to average assigned tasks

5.3.3 Varying maximum workload

Via the real-world data set, we also study the impact of maximum workload on the performance of task allocation by varying g from 4 to 8 with two increments. The corresponding results of the six algorithms under 40 tasks (i.e., 78 independent sensing measurements) are shown in Fig. 7. The maximal independent measurement of each task to be allocated is set to 3. We can found that, similar to the results in different task allocation situations, MLP has maximum average tasks per user among these testing approaches. Moreover, there is a subtle trend that the average assigned tasks per user increases with the rise of maximum workload value g. The reason for this observation lies in the fact that, when g becomes larger, each participant user can undertake more tasks. Thus the final selected user size will decrease accordingly, and the average assigned tasks per user will increase for a given set of tasks.



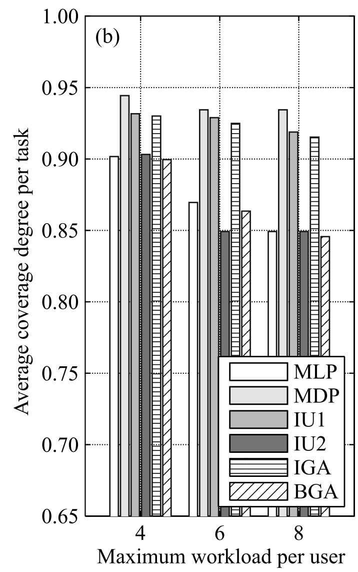


Fig. 7 Effects of maximum workload on two metrics (Real-world data set)

Moreover, for different task allocation approaches, we also study the effect of maximum workload setting on coverage degree metric. As shown in Fig. 7, the result demonstrates that maximum workload g can also influence the average coverage degree per task. Plainly, as maximum workload g increases, some user-task pair with less coverage degree will also be determined as the assignment scheme in order to extend the average assigned tasks for each user. Because MLP allocation algorithm is matched pattern length-first strategy, the influence of g will be more obvious than others. While BGA algorithm is also insensitive to the value of g. The possible reason is that participant users' trace history shows a relatively homogeneous distribution among different APs.

5.3.4 Task allocation efficiency

The allocation efficiency of these algorithms under different sensing task pool size is shown in Fig. 8. The maximum workload g is set to 8. Generally speaking, the number of allocation iteration will grow with the increase of task pool size. Among these six algorithms, MLP algorithm outperforms other assignment approaches obviously. The reason is plain enough, as explained above, MLP employs a longest matching pattern-first strategy, so it can assign more tasks to a candidate user at each iteration. While for IGA algorithm, at each iteration, it calculates all the valid user-task pairs, and determines one based on the evaluation of overall performance. Thus, the allocation iteration is equal to the tasks multiply by its corresponding independent measurement (i.e., the number of task copies). In other words, once the number of sensing tasks and independent measurement is given, the allocation iteration of IGA approach can be directly determined. In the process of user-task pair comparison, it consumes much running time.

5.3.5 Task completion ratio

Task completion are also investigated by verifying whether the allocated tasks has been performed by their chosen users. In experiments, we randomly choose independent one day to test it. If user actually passes through his allocated task's location (in WTD dataset he connect AP that the task is in), we say that he can accomplish this task. For each user, we define the *completion ratio* is the ratio of the number of completed tasks to all the tasks allocated to him. While the average task completion ratio is the mean of all selected users's task completion ratio. For different scales of released tasks, we conducted ten times of experiments (i.e., random selected ten days) to get the average value. The corresponding results are listed in Table 3. As shown in Table 3, in addition to the

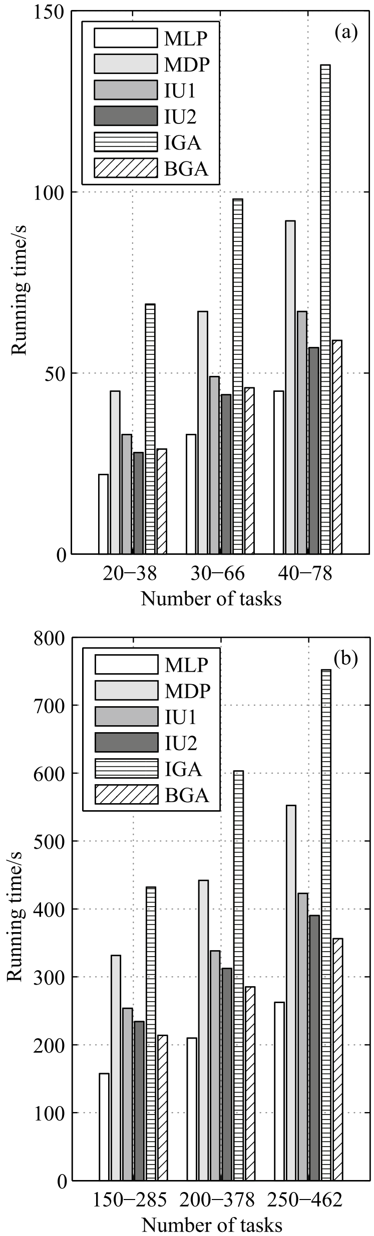


Fig. 8 The running efficiency comparison. (a) Real-world data set; (b) synthesized data set

six aforementioned methods, we also conduct the nearest-based allocation approach (i.e., NA approach) in which the tasks are assigned based on the distance between user's location and sensing task' position [32]. Basically, the performance of MDP approach is best, followed by IU1, IGA, IU2, MLP, BGA and NA algorithm. The reason is that the mobility prediction accuracy directly impact the performance of task completion. IGA and BGA approach characterize all the mobility information hidden in training dataset by Bayesian probability with the form of isolated locations, without considering the moving sequence. While pattern matching-based allocation, MDP, IU1, IU2 and MLP methods exploit mobility transfer probability in a moving sequence form (i.e., moving route), thus their prediction accuracy will be better. For nearest-based NA approach, it is insufficient to predict user's future location only based on close distance.

Table 3 Task completion ratio comparison

Algorithm	20–44	30–65	65–150	65–200
MLP	0.7404	0.776	0.7194	0.7321
MDP	0.80267	0.81267	0.7645	0.796
IU1	0.78375	0.79655	0.7392	0.745
IU2	0.7408	0.7785	0.7235	0.747
IGA	0.779	0.7843	0.7109	0.743
BGA	0.7327	0.765	0.7177	0.7433
NA	0.37634	0.3619	0.2984	0.347

6 Conclusion

In this paper, we investigate the mobile crowdsensing task optimal allocation problem in multi-task situation. To be specific, towards the goal of minimizing cost budget and maximizing sensing quality, we propose to characterize participant users' mobility regularity and make the MCS task allocation process follow users' repetitive mobility pattern. A unique characteristic of our framework is that it transforms the problem of MCS task allocation into the form of mobility pattern matching, which can be regarded as a moving route prediction problem. These two optimization goals are transformed into pattern matching length and matching support degree metrics. With respect to these two competing objectives, we devise greedy-based assignment approach to optimally match task sequence with mobility patterns discovered in advance. Extensive experiments conducted on a real-world data and a synthetic data set clearly validate the effectiveness of our proposed approaches. As for our future work, we plan to consider other factors that may affect the task allocation process, such as user interactive, social relationship and so on. Also, new approximation optimization strategies and theoretical foundations will be studied.

Acknowledgements This work was partially supported by the National Basic Research Program of China (2015CB352400), the National Natural Science Foundation of China (Grant Nos. 61402360, 61402369), the Foundation of Shaanxi Educational Committee (16JK1509). The authors are grateful to the anonymous referees for their helpful comments and suggestions.

References

1. Ganti R K, Ye F, Lei H. Mobile crowdsensing: current state and future challenges. IEEE Communications Magazine, 2011, 49(11): 32–39
2. Guo B, Wang Z, Yu Z, Wang Y, Yen N Y, Huang R, Zhou X. Mo-

bile crowd sensing and computing: the review of an emerging human-powered sensing paradigm. ACM Computing Surveys, 2015, 48(1): 7

3. Yu Z, Xu H, Yang Z, Guo B. Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints. IEEE Transactions on Human-Machine Systems, 2016, 46(1): 151–158
4. Guo B, Chen H, Yu Z, Xie X, Huangfu S, Zhang D. FlierMeet: a mobile crowdsensing system for cross-space public information reposting, tagging, and sharing. IEEE Transactions on Mobile Computing, 2015, 14(10): 2020–2033
5. Wang J, Wang Y, Zhang D, Wang L, Chen C, Lee J W, He Y. Real-time and generic queue time estimation based on mobile crowdsensing. Frontiers of Computer Science, 2017, 11(1): 49–60
6. Xiong F, Liu Y, Cheng J. Modeling and predicting opinion formation with trust propagation in online social networks. Communications in Nonlinear Science and Numerical Simulation, 2017, 44(3): 513–524
7. Wang J, Gao F, Cui P, Li C, Xiong Z. Discovering urban spatiotemporal structure from time-evolving traffic networks. In: Proceedings of the 16th Asia-Pacific Web Conference. 2014, 93–104
8. Wang J, Gu Q, Wu J, Liu G, Xiong Z. Traffic speed prediction and congestion source exploration: a deep learning method. In: Proceedings of the 16th IEEE International Conference on Data Mining. 2016, 499–508
9. Wang J, Chen C, Wu J, Xiong Z. No longer sleeping with a bomb: a duet system for protecting urban safety from dangerous goods. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017, 1673–1681
10. Thebault-Spieker J, Terveen L G, Hecht B. Avoiding the south side and the suburbs: the geography of mobile crowdsourcing markets. In: Proceedings of ACM Conference on Computer Supported Cooperative Work and Social Computing. 2015, 265–275
11. Chon Y, Lane N D, Kim Y, Zhao F, Cha H. Understanding the coverage and scalability of place-centric crowdsensing. In: Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2013, 3–12
12. Kazemi L, Shahabi C. Geocrowd: enabling query answering with spatial crowdsourcing. In: Proceedings of International Conference on Advances in Geographic Information Systems. 2012, 189–198
13. He S, Shin D H, Zhang J, Chen J, Chen J. Toward optimal allocation of location dependent tasks in crowdsensing. In: Proceedings of International Conference on Computer Communications. 2014, 745–753
14. Liu Y, Guo B, Wang Y, Wu W, Yu Z, Zhang D. TaskMe: multi-task allocation in mobile crowd sensing. In: Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2016, 403–414
15. Guo B, Liu Y, Wu W, Yu Z, Han Q. ActiveCrowd: a framework for optimized multitask allocation in mobile crowdsensing systems. IEEE Transactions on Human-Machine Systems, 2017, 47(3): 392–403
16. Feng Z, Zhu Y, Zhang Q, Ni L M, Vasilakos A V. TRAC: truthful auction for location-aware collaborative sensing in mobile crowdsourcing. In: Proceedings of International Conference on Computer Communications. 2014, 1231–1239
17. Reddy. S, Estrin D, Srivastava M. Recruitment framework for participatory sensing data collections. In: Proceedings of International Conference on Pervasive Computing. 2010, 138–155
18. Pournajaf L, Xiong L, Sunderam V. Dynamic data driven crowd sensing task assignment. Procedia Computer Science, 2014, 29(1): 1314–1323
19. Zhang D, Xiong H, Wang L, Chen G. CrowdRecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint. In: Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2014, 703–714
20. Xiong H, Zhang D, Chen G, Wang L, Gauthier V, Barnes L E. iCrowd: near-optimal task allocation for piggyback crowdsensing. IEEE Transactions on Mobile Computing, 2016, 15(8): 2010–2022
21. Hachem S, Pathak A, Issarny V. Probabilistic registration for largescale mobile participatory sensing. In: Proceedings of Pervasive Computing and Communications. 2013, 132–140
22. Kandappu T, Jaiman N, Tandriansyah R, Misra A, Cheng S F, Chen C, Lau H C, Chander D, Dasgupta K. TASKer: behavioral insights via campus-based experimental mobile crowd-sourcing. In: Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2016, 392–402
23. Kandappu T, Misra A, Cheng S F, Jaiman N, Tandriansyah R, Chen C, Lau H C, Chander D, Dasgupta K. Campus-scale mobile crowd-tasking: deployment and behavioral insights. In: Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work and Social Computing. 2016, 800–812
24. Wang L, Yu Z, Guo B, Ku T, Yi F. Moving destination prediction using sparse dataset: a mobility gradient descent approach. ACM Transactions on Knowledge Discovery from Data, 2017, 11(3): 37
25. Wang L, Hu K, Ku T, Yan X. Mining frequent trajectory pattern based on vague space partition. Knowledge-Based Systems, 2013, 50(3): 100–111
26. McNett M, Voelker G M. Access and mobility of wireless PDA users. ACM Sigmobile Mobile Computing and Communications Review, 2005, 9(2): 40–55
27. Rhee I, Shin M, Hong S, Lee K, Kim S J, Chong S. On the levy-walk nature of human mobility. IEEE/ACM transactions on networking, 2011, 19(3): 630–643
28. Srikant R, Agrawal R. Mining sequential patterns: generalizations and performance improvements. In: Proceedings of International Conference on Extending Database Technology. 1996, 1–17
29. To H, Fan L, Tran L, Shahabi C. Real-time task assignment in hyperlocal spatial crowdsourcing under budget constraints. In: Proceedings of Pervasive Computing and Communications. 2016, 1–8
30. Cheng P, Lian X, Chen Z, Fu R, Chen L, Han J, Zhao J. Reliable diversity-based spatial crowdsourcing by moving workers. Proceedings of the VLDB Endowment, 2015, 8(10): 1022–1033
31. Wang J, Wang Y, Zhang D, Wang L, Xiong H, Helal A, He Y, Wang F. Fine-grained multitask allocation for participatory sensing with a shared budget. IEEE Internet of Things Journal, 2016, 3(6): 1395–1405
32. Pournajaf L, Xiong L, Sunderam V, Goryczka S. Spatial task assignment for crowd sensing with cloaked locations. In: Proceedings of the 15th IEEE International Conference on Mobile Data Management. 2014, 73–82

Liang Wang received the PhD degree in computer science from Chinese Academy of Sciences, China. He is currently a postdoctoral researcher in Northwestern Polytechnical University, China. His research interests include mobile crowd sensing and intelligent systems.

Zhiwen Yu is currently a professor in Northwestern Polytechnical University, China. He was an Alexander Von Humboldt Fellow with Mannheim University, Germany from November 2009 to October 2010. His research interests include ubiquitous computing and HCI.

Bin Guo received the PhD degree in computer science from Keio University, Japan in 2009, He is currently a professor with Northwestern Polytechnical University, China. His research interests include ubiquitous computing, mobile crowd sensing, and HCI.

Fei Yi is currently working toward the doctoral degree at the School of Computer Science and Technology, Northwestern Polytechnical University, China. His research interests include mobile crowd sensing and intelligent systems.

Fei Xiong received the PhD degree from Beijing Jiaotong University (BJTU), China. He is currently an associate professor with BJTU. His research interests include complex networks and complex systems.