doi:10.3772/j.issn.1006-6748.2019.01.002

Dynamic data-sharing based user recruitment in mobile crowdsensing[①]

Chen Shuang(陈 爽)[②] [*] [**], Liu Min [*], Sun Sheng [*] [**], Jiao Zhenzhen [*]
([*] State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences,
Beijing 100190, P. R. China)
([**] University of Chinese Academy of Sciences, Beijing 100049, P. R. China)

Abstract

Mobile crowdsensing (MCS) has become an emerging paradigm to solve urban sensing problems by leveraging the ubiquitous sensing capabilities of the crowd. One critical issue in MCS is how to recruit users to fulfill more sensing tasks with budget restriction, while sharing data among tasks can be a credible way to improve the efficiency. The data-sharing based user recruitment problem under budget constraint in a realistic scenario is studied, where multiple tasks require homogeneous data but have various spatio-temporal execution ranges, meanwhile users suffer from uncertain future positions. The problem is formulated in a manner of probability by predicting user mobility, then a dynamic user recruitment algorithm is proposed to solve it. In the algorithm a greedy-adding-and-substitution (GAS) heuristic is repeatedly implemented by updating user mobility prediction in each time slot to gradually achieve the final solution. Extensive simulations are conducted using a real-world taxi trace dataset, and the results demonstrate that the approach can fulfill more tasks than existing methods.

Key words: mobile crowdsensing (MCS), data sharing, user recruitment, mobility prediction, dynamic decision

0 Introduction

Benefiting from the growing number of sensor-rich mobile devices (e. g. smartphones, smart vehicles, wearable devices) in urban areas, mobile crowdsensing (MCS)[1] has become an emerging paradigm to solve urban sensing problems. A typical MCS application can gather sensing requests from multiple requesters and publish them as sensing tasks, meanwhile recruit mobile users with sensing capabilities to perform tasks and pay for the sensing costs. So far, MCS has been applied in various fields, e. g. traffic monitoring[2], parking space search[3], and environment monitoring[4].

One critical issue for MCS applications is how to recruit appropriate users among the crowd to fulfill more tasks under budget constraint. Given that tasks usually have special spatial and temporal constraints, a common strategy is to recruit the nearest users to sense certain data for each task. But it is claimed that it is more economical to recruit users by implementing data sharing. That means that actually multiple tasks can reuse the same pieces of data to simultaneously satisfy their requirements by carefully selecting the sensing points. There exist many scenarios in practice where data sharing is feasible. For example, in many MCS applications such as air quality monitoring[4] or traffic monitoring[2], the sensed data usually has continuity or correlation in both spatial and temporal domains, meanwhile data requesters such as citizens may also stand reasonable data deviations in their daily lives. So requesters usually allow their tasks to be executed within a certain spatio-temporal range and consider all data sensed within the range acceptable. To this end, in case requesters require homogeneous data and the execution ranges of their tasks overlap, users can be intentionally recruited to sense data in the overlapping domains to fulfill multiple tasks simultaneously, by which the recruited users are fewer and the overall costs are undoubtedly reduced.

However, implementing efficient data sharing in user recruitment is not trivial. On one hand, various execution ranges of different tasks cause complex spatio-temporal overlapping situations, which makes different recruitment decisions deeply affect each other.

[①] Supported by the National Natural Science Foundation of China (No. 61472402, 61472404, 61732017, 61501125, 61502457).

[②] To whom correspondence should be addressed. E-mail: liumin@ict.ac.cn

Received on Feb. 25, 2018

So more comprehensive and fine-grained scheduling over the whole spatio-temporal domain is needed. But on the other hand, the users' future locations are uncertain due to the natural attribute of user mobility, which makes it difficult to judge the effectiveness of advance scheduling, thus credible data sharing becomes more challenging.

Most existing papers[5-9] either consider that the tasks are executed at abstract space-time points[5] or within non-overlapping spatio-temporal ranges[6,7], or consider that the required data is heterogeneous[8,9], all of which directly ignore the data sharing opportunity. A few papers only involve data sharing based on overlap of task execution ranges either in time dimension[10-14] or in space dimension[15], which also greatly restricts the ways to achieve data sharing. Only Ref. [16] concerns data sharing based on overlap of task execution ranges in both space and time dimensions, but it fails to consider the uncertain but valuable future mobility information of users to achieve brighter recruitment decisions.

To this end, the data-sharing based user recruitment problem under budget constraint in a realistic scenario is studied, where tasks require homogeneous data but need to be executed in various spatio-temporal ranges which may overlap, meanwhile users' future locations are uncertain. This problem is formulated in a manner of probability by predicting users' future location distributions, then a dynamic user recruitment algorithm is proposed to solve it. In the algorithm first an effective heuristic is developed to obtain a temporary solution for the problem at the beginning, then the heuristic is repeatedly applied to continually improve the solution by solving a smaller-scale problem in each following time slot. The contributions of this paper are as follows.

- The data-sharing based user recruitment problem in MCS is proposed and formalized by first considering both the spatio-temporal overlapping execution ranges of tasks and the uncertain mobility of users.
- A dynamic algorithm is proposed which repeatedly implements a greedy-adding-and-substitution heuristic by updating user mobility prediction in each time slot to continuously improve the recruitment strategy and gradually achieve the final solution.
- The algorithm with extensive simulations is evaluated using a real-world taxi trace dataset, the results prove that the approach can fulfill more tasks than existing methods.

The rest of the paper is organized as follows. In Section 1, related work is reviewed. In Section 2, the model is introduced to formulate the problem. In Section 3, the algorithm details are elaborated. The simulation results are showed in Section 4 and the paper is concluded in Section 5.

1 Related work

A number of papers have studied user recruitment or task assignment problems in MCS. A majority of these papers[5-9] do consider the spatial and temporal constraints of tasks. But they either assume that tasks are executed at abstract space-time points[5] or within spatio-temporal ranges[6,7], or focus on heterogeneous data requirements for different tasks[8,9], all of which do not involve data sharing at all.

Only a few papers involve data sharing in MCS by considering that task execution ranges overlap either in time dimension[10-14] or in space dimension[15]. Ref. [10] designed the optimal transmission schedule for a mobile user in MCS to make a tradeoff between the amount of data transmitted and the energy consumption by sharing data among requests with different durations. Ref. [11] studied the optimal sampling time for multiple time-sensitive tasks in a smartphone by reusing data to minimize the energy consumption while ensuring sensing quality. But Refs[10,11] considered task scheduling on a single device where only temporal overlapping execution ranges were concerned and data was shared in time dimension. Ref. [12] presented a task allocation approach for multiple users by sharing sensing services with different tasks to achieve both fairness and energy efficiency, but it only utilized the overlapping sensing intervals of tasks while assuming the sensing areas are the same. Ref. [13] considered a dynamic participant recruitment problem to minimize the sensing cost while maintaining certain level of probabilistic coverage. Ref. [14] proposed an participant selection scheme by introducing caching into MCS to store data for future tasks. But in Refs[13,14], the tasks were restricted to be executed in non-overlapping Points of Interest (PoIs), thus data could only be shared among tasks in the same location. Besides, Ref. [15] studied the quality aware sensing coverage problem under budget constraint by dealing with the overlapping execution areas of tasks, but it assumed that the temporal requirements were irrelevant. All these papers simplify the overlapping scenarios and limit the possible ways to achieve data sharing.

Ref. [16] considered data sharing based on overlap of task execution ranges in both space and time dimensions. It proposed task assignment methods in two scenarios of fixed budget constraint in each time slot and total budget constraint over the entire campaign, but it only applied local heuristics without leveraging users' future mobility information. However, this information is vital for intelligent data-sharing based user recruitment.

2 Model and problem formation

2.1 System model

A typical MCS system is considered consisting of a platform, multiple data requesters and multiple mobile users. The platform receives sensing tasks from data requesters, and recruits mobile users to perform tasks with certain costs. A homogeneous MCS process in area $\mathcal{L}$ and period $\mathcal{T}$ is focused. Without loss of generality, the target area is equally divided into L grids and the whole period is divided into T time slots to generate $L \times T$ spatio-temporal grid cells. The precision depends on specific application requirement. It is denoted the set of tasks as $S = \{1, 2, \cdots, S\}$. For each task i, $\mathcal{SR}_i$ is denoted as the spatial execution range, which is formalized as

$$\mathcal{SR}_i = \{ l \mid \| Co(l) - Co(l_i) \|_2 \leqslant R_i \} \tag{1}$$

where $Co(l)$ is the coordinate of location l. Eq. (1) means $\mathcal{SR}_i$ is a circle centered on location l_i and with a radius of R_i. $\mathcal{TR}_i$ is also denoted as the temporal execution range, which is formalized as

$$\mathcal{TR}_i = \{ t \mid 0 \leqslant t - t_i \leqslant D_i \} \tag{2}$$

which means $\mathcal{TR}_i$ starts in time slot t_i and lasts a duration of D_i. Note that based on personalized precision requirements, R_i and D_i can be diverse for different tasks. It is considered that all users are reliable and can update qualified data, which can be achieved by applying reputation system[17] or truth estimation technology[18] to identify and screen out reliable users beforehand, and the details are omit here for simplicity. Then a task can be fulfilled if at least one user is recruited to collect data within its both spatial and temporal execution ranges. Besides, it is also considered all task requirements are known at the beginning of the MCS process, which is rational when the sensing period is not too long and task reservation is available.

The set of users are denoted as $\mathcal{U} = \{1, 2, \cdots, U\}$, who can move in and out the target area. The users' future locations are uncertain due to mobility. Note that many applications can record users' historical trajectories such as by periodically reading the GPS information, one can leverage the historical trajectories to predict users' future locations to some extent. For simplicity, it is assumed that a user stays at a single location in each time slot and it is considered he stays at location $L + 1$ if he is outside the target area. In order to generally characterize the uncertainty of users' locations, it is considered any user j in any time slot t follows a distribution of $\theta_j(t) = (\theta_j(1, t), \theta_j(2, t), ..., \theta_j(L + 1, t))$, which means user j will appear in location l in time slot t with a probability of $\theta_j(l, t)$. It is also denoted $\boldsymbol{\theta}(t) = (\theta_1(t), \theta_2(t), ..., \theta_U(t))$. Next, the widely used one-order Markov model is adopted to make the prediction similar to Refs[8,19]. Note that more complex prediction methods such as Bayesian learning[20] can be easily extended and they are omit here. The probability p_{mn} is calculated that users move from location m to location n within adjacent time slots based on historical trajectories, then form the location transition matrix as

$$\boldsymbol{P} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,L+1} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,L+1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{L+1,1} & p_{L+1,2} & \cdots & p_{L+1,L+1} \end{bmatrix} \tag{3}$$

It is assumed that users' exact locations can be obtained at the beginning of each time slot, then the location distribution of user j for any future time slot t can be predicted as

$$\boldsymbol{\theta}(t) = \boldsymbol{\theta}(t_0) \boldsymbol{P}^{t - t_0}, \tag{4}$$

where, t_0 is the current time slot and $\boldsymbol{\theta}(t_0)$ represents users' newly arrived locations in time slot t_0. $\theta_j(l, t_0) = 1$ if user j does stay at location l in time slot t_0 and $\theta_j(l, t_0) = 0$ if not. In this way, in current time slot t_0, users' location distributions for all future time slots can be expressed as

$$\boldsymbol{\theta}^{(t_0)} = (\boldsymbol{\theta}(t_0), \boldsymbol{\theta}(t_0 + 1), ..., \boldsymbol{\theta}(T)) \tag{5}$$

From Eq. (4) and Eq. (5) it can be seen, location prediction $\boldsymbol{\theta}(t_0)$ for future time slot t is more accurate if t is closer to current time slot t_0 and vice versa. This property will be exploited in subsequent algorithm design.

2.2 Problem formulation

Given the set of tasks $\mathcal{S}$ with spatia-temporal execution ranges and the set of users $\mathcal{U}$ with future location prediction, the data-sharing based user recruitment problem aims at maximizing the total number of fulfilled tasks under budget constraint. It is considered that a user is recruited once if he is selected to sense a regular amount of data in a time slot at the location he stays. A user can be recruited many times in different time slots. So the solution space of the user recruitment problem is the set of all user-time combinations $\mathcal{C} = \{(j, t) \mid j \in \mathcal{U}, t \in \mathcal{T}\}$. The final solution is to select a subset of user-time combinations $\mathcal{C}_r \subseteq \mathcal{C}$. A user-time combination (j, t) is $\in \mathcal{C}_r$ if user j is selected in time slot t. As the scenario is focused on where the sensed data is homogeneous, and it is considered the recruitment cost is fixed for all users, so budget B represents the maximum recruitment times, which satisfies:

$$|\mathcal{C}_r| \leqslant B \qquad (6)$$

On the other hand, as users' future locations are uncertain, investigating task fulfillment in a probabilistic way is aimed. The fulfillment probability $FP_i(j,t)$ of task i generated by a single user-time combination (j,t) can be expressed as

$$FP_i(j,t) = \sum_{l \in S\mathcal{R}_i} \theta_j(t,l)\, \mathbb{T}(t \in \mathcal{TR}_i) \qquad (7)$$

where, $\mathbb{T}(t \in \mathcal{TR}_i) = 1$ if $t \in \mathcal{TR}_i$ and $\mathbb{T}(t \in \mathcal{TR}_i) = 0$ if $t \notin \mathcal{TR}_i$. $FP_i(j,t)$ is actually the probability that user j appears in the spatial execution range $\mathcal{SR}_i$ of task i in time slot t, in case that t is within the temporal execution range $\mathcal{TR}_i$ of task i. Thus, given the entire recruitment solution $\mathcal{C}_r$, the fulfillment probability of task i can be calculated as

$$R_i(\mathcal{C}_r) = 1 - \prod_{(j,t) \in \mathcal{C}_r} (1 - FP_i(j,t)) \qquad (8)$$

which is the joint probability contributed by all user-time combinations in $\mathcal{C}_r$. So the expected total number of fulfilled tasks for task set $\mathcal{S}$ can be calculated as

$$R_S(\mathcal{C}_r) = \sum_{i \in S} R_i(\mathcal{C}_r) \qquad (9)$$

Finally, the probabilistic data-sharing based user recruitment optimization problem is formulized with the object of maximizing the expected total number of fulfilled tasks under budget constraint as follows.

$$\begin{aligned} &\max\ R_S(\mathcal{C}_r) \\ &\text{s.t.} \quad |\mathcal{C}_r| \leqslant B. \end{aligned} \qquad (10)$$

This problem is NP-hard. Eq. (10) can be degenerated to a special case where the total sensing period is one time slot and all users' locations are determined, then the problem directly becomes the maximum coverage problem, which is well known as NP-complete. In the next section, it is aimed to seek alternative heuristics to solve this problem.

3 Algorithm details

3.1 Algorithm description

In order to solve the probabilistic data-sharing based user recruitment optimization problem, the main idea is to develop polynomial time heuristics while fully leverage users' future location information. To this end, a dynamic user-time combination selection algorithm is proposed. In this algorithm, a greedy-adding-and-substitution (GAS) heuristic is first developed to obtain a temporary solution for the complete problem in the initial time slot, then this heuristic is repeatedly implemented to solve a smaller-scale problem to improve the solution in each following time slot with the help of updating user location prediction. Below the GAS heuristic will be elaborated first, which is the building block of our algorithm, then the complete dynamic algorithm is given.

3.1.1 GAS heuristic

The greedy-adding-and-substitution (GAS) heuristic is a general method for a wide class of covering problems[21], which is modified to apply into the problem, the details are showed in Algorithm 1. Without loss of generality, time slot t is taken as an example. In which, given the unfulfilled task set $\mathcal{S}^{(t)}$, residual budget $B^{(t)}$ and newly updated user location prediction $\boldsymbol{\theta}^{(t)}$, the GAS heuristic aims to solve a smaller-scare probabilistic user recruitment problem by selecting a subset of user-time combinations $\mathcal{C}_r^{(t)}$ from current candidate combination set $\mathcal{C}^{(t)}$ to maximize the expected total number of fulfilled tasks. The current candidate combination set $\mathcal{C}^{(t)}$ contains all combinations that can be selected from current time slot t to the last time slot, which is

$$\mathcal{C}^{(t)} = \{(j',t') \mid j' \in \mathcal{U}, t' \geqslant t\} \qquad (11)$$

The GAS heuristic solves the problem in an iterative way. In each iteration, a greedy adding phase is implemented followed by a substitution phase. In the greedy adding phase, an unselected user-time combination is chosen which maximizes the marginal contribution to the expected total number of fulfilled tasks and is added into the selected user-time combination set $\mathcal{C}_r^{(t)}$ in line 5-10. According to Eq. (9), the marginal contribution of adding a user-time combination (j_0,t_0) into the selected combination set $\mathcal{C}_r$ given task set $\mathcal{S}$ can be calculated as

$$\begin{aligned} \Delta R_{S,C_r}(j_0,t_0) &= R(\mathcal{C}_r \cap (j_0,t_0)) - R \\ &= \sum_{i \in \mathcal{S}} \prod_{(j,t) \in \mathcal{C}_r} (1 - FP_i(j,t)) FP_i(j_0,t_0) \end{aligned} \qquad (12)$$

Algorithm 1 GAS heuristic

```
Input: current unfulfilled task set $\mathcal{S}^{(t)}$, current candidate us-
  er-time combination set $\mathcal{C}^{(t)}$ with location prediction $\boldsymbol{\theta}^{(t)}$,
  residual budget $B^{(t)}$.
Output: temporary recruitment solution $\mathcal{C}_r^{(t)}$.
1.  $\bar{B}^{(t)} \leftarrow B^{(t)}$, $\bar{\mathcal{C}}^{(t)} \leftarrow \mathcal{C}^{(t)}$;
2.  $\mathcal{C}_r^{(t)} \leftarrow \varnothing$;
3.  while $\bar{B}^{(t)} > 0$ and $\mathcal{C}^{(t)} \neq \varnothing$ do
4.      // Greedy adding phase
5.      for all $(j,t) \in \bar{\mathcal{C}}^{(t)}$ do
6.          Calculate $\triangle R_{\mathcal{S}^{(t)},\mathcal{C}_r^{(t)}}(j, t)$ based on Eq. (12) as-
            suming adding $(j,t)$ into $\mathcal{C}_r^{(t)}$;
7.      end for
8.      Select $(j^*,t^*) \in \bar{\mathcal{C}}^{(t)}$ which leads to the maximum
        $\triangle R_{\mathcal{S}^{(t)},\mathcal{C}_r^{(t)}}(j, t)$;
```

```
9.  $\mathcal{C}_r^{(t)} \leftarrow \mathcal{C}_r^{(t)} \cup (j^*, t^*)$, $\bar{\mathcal{C}}^{(t)} \leftarrow \bar{\mathcal{C}}^{(t)} / (j^*, t^*)$;
10. $\bar{B}^{(t)} \leftarrow \bar{B}^{(t)} - 1$;
11. // Substitution phase;
12. for all $(j, t) \in \mathcal{C}_r^{(t)}$ do
13.     for all $(j', t') \in \bar{\mathcal{C}}^{(t)}$ do
14.         Calculate $\mathcal{R}_{S(t)}(\mathcal{C}_r^{(t)} \cup (j', t')/(j, t))$ based on
            Eq. (9) assuming replacing $(j, t)$ with $(j', t')$;
15.     end for
16.     Select $(j^*, t^*) \in \bar{\mathcal{C}}^{(t)}$ which leads to the maximum
        $R_{S(t)}(\mathcal{C}_r^{(t)} \cup (j', t')/(j, t))$;
17.         if $R_{\mathcal{S}(t)}(C_r^{(t)} \cup (j^*, t^*)/(j, t)) > R_{\mathcal{S}(t)}(\mathcal{C}_r^{(t)})$
            then
18.         Record $(j^*, t^*)$ as substitution object of $(j, t)$;
19.         end if
20.  end for
21. Select $(j^{**}, t^{**})$ among all substitution objects which
    leads to the maximum $R_{\mathcal{S}(t)}(\mathcal{C}_r^{(t)} \cup (j', t')/(j, t))$;
22. $\mathcal{C}_r^{(t)} \leftarrow \mathcal{C}_r^{(t)} \cup (j^{**}, t^{**})$, $\bar{\mathcal{C}}^{(t)} \leftarrow \bar{\mathcal{C}}^{(t)} / (j^{**}, t^{**})$;
23. $\mathcal{C}_r^{(t)} \leftarrow \mathcal{C}_r^{(t)} / (j, t)$, $\bar{\mathcal{C}}^{(t)} \leftarrow \bar{\mathcal{C}}^{(t)} \cup (j, t)$;
24. end while
35. return $\mathcal{C}_r^{(t)}$;
```

Greedy adding is a simple myopic method, and the decision is irreversible once a combination is selected even if finding it unsuitable later, which can not guarantee the optimization. So a substitution phase is also implemented to further improve the result by properly adjusting previous selected combinations. Elaborately speaking, a selected user-time combination is replaced with an unselected candidate, in case the substitution can improve the result to the most extent. For each selected user-time combination (j,t), first the corresponding expected number of fulfilled tasks $\Delta R_{\mathcal{S}(t)}(\mathcal{C}_r^{(t)} \cup (j', t')/(j,t))$ is calculated by assuming (j,t) is replaced with any unselected candidate (j', t') in line 13-15. Next the unselected candidate is selected which leads to the maximum expected number of fulfilled tasks in Line 16, and this candidate is recorded as the substitution object of (j,t) if it can achieve a better result compared with (j,t) in line 17-19. Finally the unselected candidate among all the substitution objects of selected combinations is selected which leads to the best result and does make the substitution in line 21-23. Note that at most one selected combination in each iteration of the GAS heuristic is replaced only for the sake of algorithm complexity. In an extreme case, if all selected combinations are allowed to be replaced, it just becomes the exhaustive search, which should be avoided. The GAS heuristic ends when the residual budget becomes zero or the current candidate combination set becomes empty, then outputs the temporary solution $\mathcal{C}_r^{(t)}$ in line 25.

3.1.2 Complete dynamic algorithm

The complete dynamic algorithm is displayed in Algorithm 2. Note that if the GAS heuristic is only implemented once in the initial time slot, the solution may not be good enough, as users' location prediction may not be accurate for distant time slots. According to Eq. (4) and Eq. (5), the accuracy of future location prediction $\boldsymbol{\theta}^{(t_0)}$ can be gradually improved based on users' newly arrived locations as the process goes on. So the solution of the GAS heuristic in the initial time slot is a temporary one, and the GAS heuristic is repeatedly called in each following slot with the help of updated user location prediction $\boldsymbol{\theta}^{(t)}$ to improve the solution in line 4-5. For the temporary solution in each time slot t, user-time combinations of $A\mathcal{C}_r^{(t)}$ is only recruited that can execute tasks exactly in that time slot in line 6, then they are added into the final recruitment solution $\mathcal{C}_r$ in line 7. $A\mathcal{C}_r^{(t)}$ is denoted as

$$A\mathcal{C}_r^{(t)} = \{(j', t') \mid (j', t') \in \mathcal{C}_r^{(t)}, t' = t\} \tag{13}$$

After tasks are executed, the residual budget and user location prediction for the next time slot in line 8-9 are updated. The complete dynamic algorithm continuously runs until the MCS process goes to the end, or all tasks are fulfilled, or the budget is exhausted. Finally the recruitment solution $\mathcal{C}_r$ is outputted in line 12, which consists of $A\mathcal{C}_r^{(t)}$ in all time slots.

Algorithm 2 Complete dynamic algorithm

```
Input: total task set $\mathcal{S}$, total user-time combination set , total
  budget $B$, users' historical trajectories.
Output: final recruitment solution $\mathcal{C}_r$.
1.  $t = 1$;
2.  $\mathcal{S}^{(t)} \leftarrow \mathcal{S}$, $\bar{B}^{(t)} \leftarrow B$;
3.  while $t \leqslant T$ and $\mathcal{S}^{(t)} \neq \emptyset$ and $B^{(t)} > 0$ do
4.      Update $\boldsymbol{\theta}^{(t)}$ according to Eq. (5);
5.      Run GAS Heuristic to get $\mathcal{C}_r^{(t)}$;
6.      Recruit $A\mathcal{C}_r^{(t)}$ based on Eq. (13) to execute tasks in
        time slot $t$ and update $\mathcal{S}^{t+1}$;
7.      $\mathcal{C}_r = \mathcal{C}_r \cup A\mathcal{C}_r^{(t)}$;
8.      $\mathcal{C}^{(t+1)} \leftarrow \mathcal{C}^{(t)} / \{(j', t') \mid j' \in \mathcal{U}, t' = t\}$;
9.      $B^{(t+1)} \leftarrow B^{(t)} - |A\mathcal{C}_r^{(t)}|$;
10.   $t \leftarrow t + 1$;
11.  end while
12.  return $\mathcal{C}_r$;
```

3.2 Complexity analysis

For the GAS heuristic, the computational complexity is dominated by line 14, which runs at most $B \times B \times UT$ times due to the iterations in line 3, 12 and

13. Besides, the calculation in line 14 requires at most $S \times B \times |\mathcal{SR}|$ operations according to Eqs(7-9), $|\mathcal{SR}|$ is the upper number of locations that are within the spatial execution range of a task, which is based on Eq. (1). So the computational complexity of the GAS heuristic is $O(B^3 TUS|\mathcal{SR}|)$. Finally, the complete dynamic algorithm runs the GAS heuristic for at most T times, thus the total computational complexity is $O(B^3 T^2 US|\mathcal{SR}|)$, which is polynomial time.

4 Simulation results

4.1 Simulation setup

In the simulations, the widely used real-world roma/taxi dataset[22] is adopted as user mobility traces. The roma/taxi dataset contains periodically collected GPS coordinates of approximately 320 taxi cabs in Rome, Italy over 30 days in 2014. An 11.1km × 16.6km rectangle area in (12.4° – 12.6°E, 41.85° – 41.95°N) is chosen as the target area, which is in downtown and has the highest taxi density. The target area is divided into 20 × 30 grids, and a time slot as 5 minutes is considered. The sensing period of each MCS process is to begin at 8:00 a.m. in any random day from the second week to the last day of traces. Taxis staying the longest in the target area are selected as users, and transition matrix P based on users' trajectories is calculated in the same period of time as the sensing period during the past week. Tasks are generated with spatial execution range $\mathcal{SR}_i$ and temporal execution range $\mathcal{TR}_i$. For simplicity, the radius R_i of $\mathcal{SR}_i$ is set as well as the duration D_i of $\mathcal{TR}_i$ for each task be the same. Besides, $\mathcal{SR}_i$ and $\mathcal{TR}_i$ of all tasks are considered independently and uniformly to be distributed within the target spatio-temporal domain. The default parameters are listed in Table 1.

Table 1 default parameters

Number of Users	Number of Tasks	Budget	Radius	Duration	Period
100	200	50	3	3	30

Several existing methods are also implemented for comparison. As mentioned above only Ref. [16] involves data sharing based on overlap of task execution ranges in both spatial and temporal dimensions, besides it also studies the user recruitment problem with budget constraint. Thus, three existing methods developed in Ref. [16] are chosen called AdaptB, AdaptT and AdaptS for comparison. Briefly speaking, AdaptB tries to decide whether to recruit an arriving user or not in real time using an ε-greedy algorithm based on the user's priority index and the residual budget. The user's priority index is the number of tasks that the user can newly fulfill at current time. AdaptS and AdaptT modify the priority index by further taking the area popularity and task urgency into consideration respectively. A random algorithm is also employed as the benchmark, which chooses B random user-time combinations. The total number of fulfilled tasks is the key metric for the performance.

Simulations are conducted under a wide range of settings by varying each parameter. In simulation of varying the number of users, as there are basically no more than 150 taxis in the target area during the sensing period, more traces are simply added in the same period of time as the sensing period in following days to achieve up to 300 users. All results are averaged by running simulations multiple times in various days.

4.2 Performance evaluation

Fig. 1 shows the proposed algorithm always performs the best as the number of users varies, the AdaptT, AdaptS and AdaptB algorithms perform worse in turn, and the random algorithm performs the worst. As the number of users increases, the numbers of fulfilled tasks for all algorithms except the Random rise because more suitable users are available and more efficient recruitment can be achieved. In Fig. 2, the algorithm also outperforms others by varying the number of tasks. The number of fulfilled tasks increases almost linearly for all algorithms as the number of tasks increases, because adding more tasks in the fixed spatio-temporal domain brings more overlapping requirements and data sharing can be better achieved. Fig. 3 shows the proposed algorithm is more advantageous when the budget is tight, which indicates that the algorithm operates in a more frugal way. The results become closer when the budget increases as all algorithms can select adequate

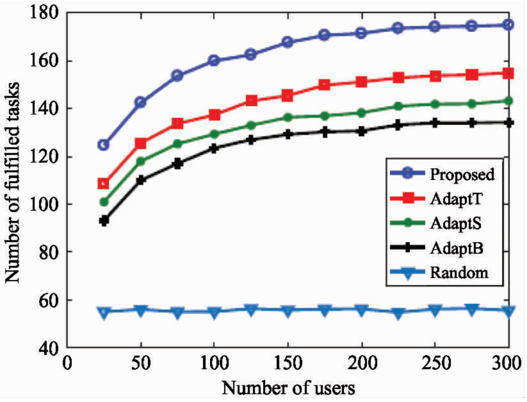

Fig. 1 Number of total fulfilled tasks vs. number of users

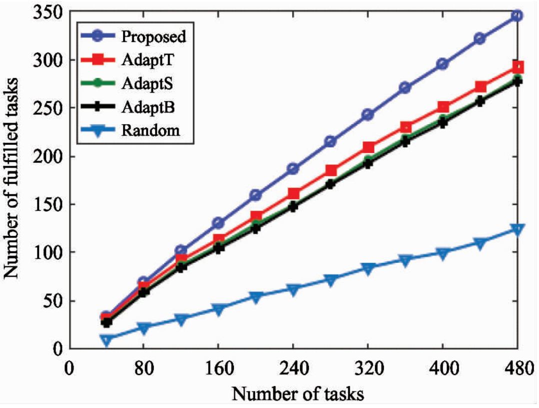

Fig. 2 Number of total fulfilled tasks vs. number of tasks

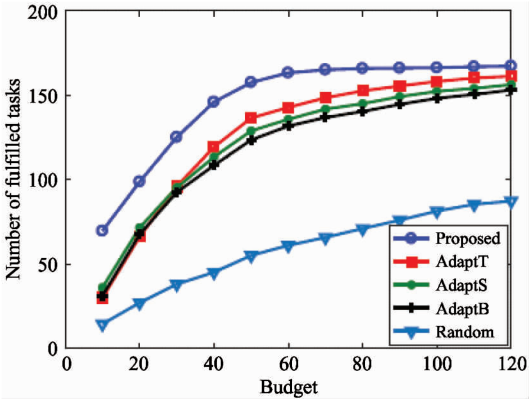

Fig. 3 Number of total fulfilled tasks vs. budget

usertime combinations without worrying about budget shortfalls.

Fig. 4 and Fig. 5 display the effects of different radiuses R_i and durations D_i of task requirements on the results. As expected, the results rise when radius and duration increase because tasks with more relaxed requirements can be satisfied more easily for all algorithms. This algorithm is more advantageous when radius and duration are smaller, indicating that the algorithm can match task requirements more intelligently.

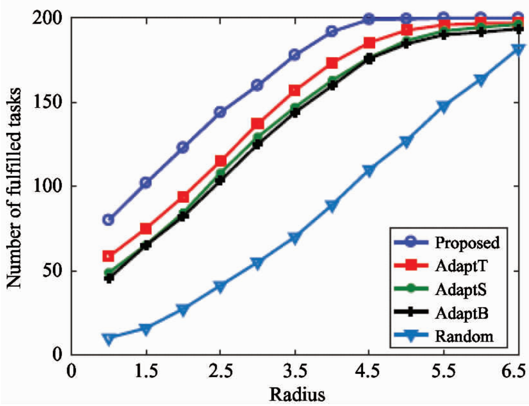

Fig. 4 Number of total fulfilled tasks vs. radius

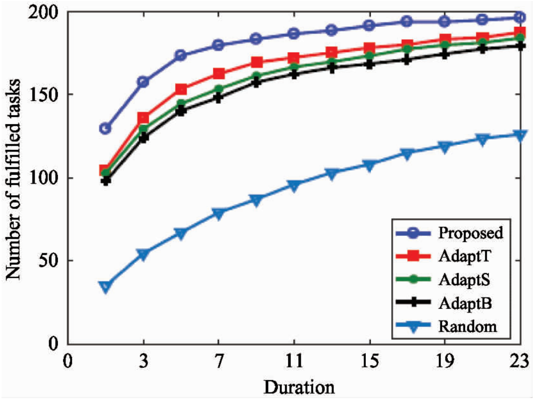

Fig. 5 Number of total fulfilled tasks vs. duration

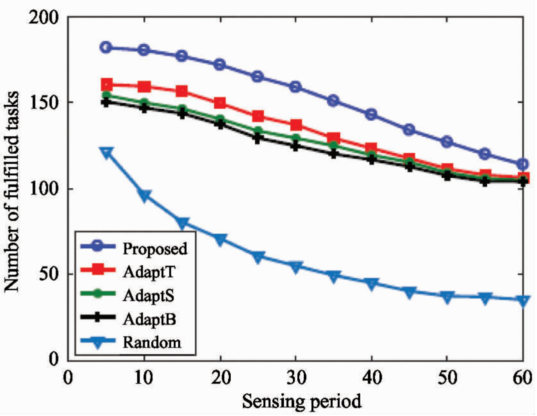

Fig. 6 Number of total fulfilled tasks vs. sensing period

In Fig. 6, the results decrease for all algorithms as the sensing period extends, because tasks are distributed more dispersedly and data sharing becomes more difficult. This algorithm still keeps superior to others, but it is noticed that the performance is more competitive when the sensing period is not too long.

To further investigate the effects of sensing period on the results without interference factors, the period time is varied in proportion with the number of tasks and the budget based on the default values. The results are showed in Fig. 7 and Fig. 8, where Fig. 8 transforms the number of fulfilled tasks in Fig. 7 to the fulfillment ratio by dividing the total task number. The results show that the algorithm performs better with a considerably long sensing period but the advantage is gradually diminished as the sensing period extends. This is rational, because the performance of the proposed algorithm directly relies on the accuracy of users' future location prediction. When the sensing period becomes longer, the overall prediction accuracy will inevitably decline, yet which can be overcome in practice, as the length of sensing period of each MCS process can be controlled and processes are launched more frequently. Besides, more powerful location prediction methods such as collecting the speed for vehicles or analysing context information for pedestrians can also be applied. There should be a tradeoff between the algorithm performance and practical development difficulty, details is left for future work.

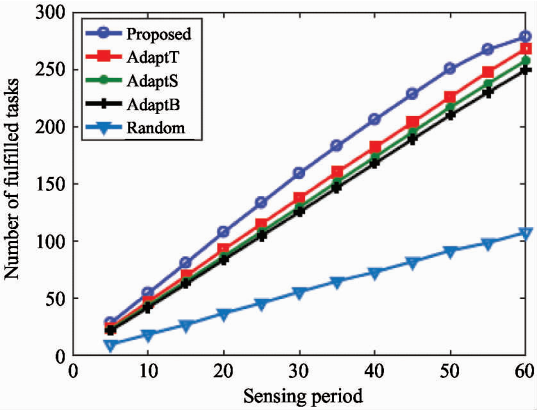

Fig. 7 Number of fulfilled tasks vs. sensing period (in proportion with number of tasks and budget)

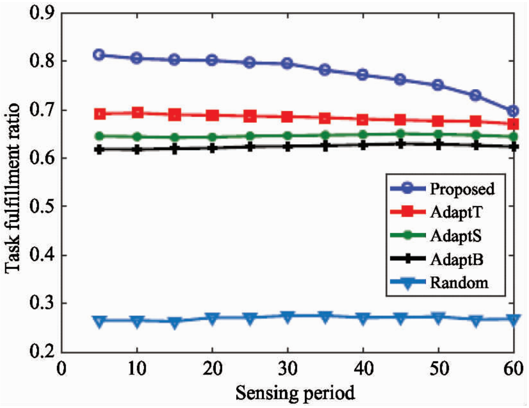

Fig. 8 Task fulfillment ratio vs. sensing period (in proportion with number of tasks and budget)

5 Conclusion

This paper studies the data-sharing based user recruitment problem under budget constraint by first considering both overlapping spatio-temporal requirements for tasks and the mobility prediction for users. A novel dynamic user recruitment algorithm is proposed by repeatedly applying a GAS heuristic with updated user mobility prediction to gradually achieve the solution. Extensive simulations demonstrate that this approach is superior to existing methods.

References

[1] Ganti R K, Ye F, Lei H. Mobile crowdsensing: Current state and future challenges [J]. *IEEE Communication Magazine*, 2011, 49(11): 32-39

[2] Thiagarajan A, Ravindranath L, LaCurts K, et al. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones[C]. In: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, Berkeley, USA, 2009. 85-98

[3] Nawaz S, Efstratiou C, Mascolo C. Parksense: A smartphone based sensing system for on-street parking[C]. In: Proceedings of the 19th ACM Annual International Conference on Mobile Computing & Networking, Miami, USA, 2013. 75-86

[4] Dutta P, Aoki P M, Kumar N, et al. Common sense: participatory urban sensing using a network of handheld air quality monitors[C]. In: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, Berkeley, USA, 2009. 349-350

[5] Cheung M H, Southwell R, Hou F, et al. Distributed time-sensitive task selection in mobile crowdsensing[C]. In: Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Hangzhou, China, 2015. 157-16

[6] Zhang D, Xiong H, Wang L, et al. Crowdrecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint[C]. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Seattle, USA, 2014. 703-714

[7] Xiong H, Zhang D, Chen G, et al. Crowdtasker: maximizing coverage quality in piggyback crowdsensing under budget constraint[C]. In: Proceedings of the 2015 IEEE International Conference on Pervasive Computing and Communications, St. Louis, USA, 2015. 55 - 62

[8] Song Z, Liu C H, Wu J, et al. Qoi-aware multitaskoriented dynamic participant selection with budget constraints[J]. *IEEE Transactions on Vehicular Technology*, 2014, 63(9): 4618-4632

[9] Wang J, Wang Y, Zhang D, et al. Fine-grained multitask allocation for participatory sensing with a shared budget[J]. *IEEE Internet of Things Journal*, 2016, 3(6): 1395-1405

[10] Wu W, Wang J, Li M, et al. Energy-efficient transmission with data sharing in participatory sensing systems [J]. *IEEE Journal on Selected Areas in Communications*, 2016, 34(12): 4048-4062

[11] Wang J, Tang J, Xue G, et al. Towards energy-efficient task scheduling on smartphones in mobile crowd sensing systems[J]. *Elsevier Computer Networks*, 2017, 115: 100-109

[12] Zhao Q, Zhu Y, Zhu H, et al. Fair energy-efficient sensing task allocation in participatory sensing with smartphones[C]. In: Proceedings of the 33th Annual IEEE International Conference on Computer Communications, Toronto, Canada, 2014. 1366-1374

[13] Li H, Li T, Wang Y. Dynamic participant recruitment of mobile crowd sensing for heterogeneous sensing tasks [C]. In: Proceedings of the 12th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, Dallas, USA, 2015. 144

[14] Li H, Li T, Li F, et al. Enhancing participant selection

through caching in mobile crowd sensing[C]. In: Proceedings of the 24th IEEE/ACM International Symposium on Quality of Service, Beijing, China, 2016. 1-10

[15] Zhang M, Yang P, Tian C, et al. Quality-aware sensing coverage in budget-constrained mobile crowdsensing networks[J]. *IEEE Transactions on Vehicular Technology*, 2016, 65(9): 7698-7707

[16] To H, Fan L, Tran L, et al. Real-time task assignment in hyperlocal spatial crowdsourcing under budget constraints[C]. In: Proceedings of the 2016 IEEE International Conference on Pervasive Computing and Communications, Sydney, Australia, 2016. 1-8

[17] Wang X, Chen W, Mohapatra P, et al. Artsense: Anonymous reputation and trust in participatory sensing[C]. In: Proceedings of the 32th Annual IEEE International Conference on Computer Communications, Turin, Italy, 2013. 2517-2525

[18] Peng D, Wu F, Chen G. Pay as how well you do: A quality based incentive mechanism for crowdsensing[C]. In: Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Hangzhou, China, 2015. 177-186

[19] Liu X, Karimi H A. Location awareness through trajectory prediction[J]. *Computers, Environment and Urban Systems*, 2006, 30(6): 741-756

[20] Anagnostopoulos T, Anagnostopoulos C, Hadjiefthymiades S, et al. Predicting the location of mobile users: a machine learning approach[C]. In: Proceedings of the 2009 International Conference on Pervasive Services, London, UK, 2009. 65-72

[21] Daskin M, Network and discrete location: models, algorithms, and applications[M]. John Wiley & Sons, 2011. 92-153

[22] Bracciale L, Bonola M, Loreti P, et al. Crawdad dataset roma/taxi [EB/OL]. http://crawdad. org/roma/taxi/20140717: Crawdad, 2014

Chen Shuang, born in 1991. He is currently pursuing the Ph. D degree at the Networking Technology Research Centre, Institute of Computing Technology, Chinese Academy of Sciences. He received his B. S. degree in electronic information engineering from Huazhong University of Science and Technology, China in 2012. His research interests include mobile crowdsensing and Internet of Things.