



Task Assignment for Semi-opportunistic Mobile Crowdsensing

Wei Gong¹, Baoxian Zhang^{1(✉)}, and Cheng Li^{2,3}

¹ Research Center of Ubiquitous Sensor Networks,
University of Chinese Academy of Sciences, Beijing 100049, China
gongwei11@mails.ucas.ac.cn, bxzhang@ucas.ac.cn

² School of Computer and Information Engineering, Tianjin Chengjian University,
Tianjin 300384, China

³ Faculty of Engineering and Applied Science, Memorial University,
St. John's, NL A1B 3X5, Canada
licheng@mun.ca

Abstract. In this paper, we propose a novel crowdsensing paradigm called semi-opportunistic sensing, which is aimed to achieve high task quality with low human involvement. In this paradigm, each mobile user can provide multiple path choices to reach her destination, which largely broadens the task assignment space. We formulate the task assignment problem in this paradigm of maximizing total task quality under incentive budget constraint and user travel time constraints. We prove this problem is NP-hard and then propose two efficient heuristic algorithms. First, we propose a Best Path/Task first algorithm (BPT) which always chooses current best path and current best task into the assignment list. Second, we propose an LP-relaxation based algorithm (LPR), which greedily assigns paths and tasks with the largest values in LP relaxation solution. We deduce the computational complexities of the proposed algorithms. We evaluate the performance of our algorithms using real-world traces. Simulation results show that our proposed crowdsensing paradigm can largely increase overall task quality compared with the opportunistic sensing paradigm where each user has only one fixed path. Simulation results also show that our proposed algorithms are efficient and their performance is close to the optimal solution.

Keywords: Mobile crowdsensing · Task assignment
Semi-opportunistic sensing

1 Introduction

Recently, mobile crowdsensing [2] has become a cost-effective way to collect massive sensing data and thus enabled tremendous real-world applications such

This work was supported in part by the NSF of China under Grant Nos. 61531006, 61471339, 61872331, the Natural Sciences and Engineering Research Council (NSERC) of Canada (Discovery Grant RGPIN-2018-03792), and the InnovateNL SensorTECH Grant 5404-2061-101.

as environment monitoring, traffic condition monitoring, and smart city [3,7,8,12]. In a mobile crowdsensing system, the service platform recruits mobile users to perform tasks by exploiting the sensing and computing ability of smart devices at the users.

Based on whether tasks are performed consciously by users, existing work for mobile crowdsensing can be categorized into two types: opportunistic sensing and participatory sensing. In opportunistic sensing, users move following their daily routines and complete sensing tasks with minimum disturbance. Sensing data along their moving paths can be collected automatically without human involvement and thus leads to low employment expenses of opportunistic sensing users. However, since the paths taken by opportunistic sensing users are predetermined based on users' moving patterns, tasks not on any mobile user's path will not be performed. This often results in limited task coverage. In contrast, the moving paths of participatory sensing users are scheduled by the service platform. Service platform can ask such users to move to any task location within their travel capability to perform tasks at the cost of high employment expenses.

Task assignment [4] is one of the most important problems for mobile crowdsensing. In participatory sensing, [9] aims to minimize the task expiration penalty by user selection and path planning, while [5] studied the problem of user selection and path planning in an online manner to maximize total task quality under given travel budget. In opportunistic crowdsensing, [10] studied the multi-objective optimization problem of maximizing total task quality under energy consumption constraints of users' smart devices. [11] studied the problem of maximizing total task quality under incentive budget constraint and proposed an approximation algorithm exploiting the submodular property of the used utility function. High task quality and low employment expenses are two crucial goals in the design of effective task assignment mechanisms for mobile crowdsensing. However, the above two crowdsensing paradigms cannot meet these two goals simultaneously. In this paper, we propose a new mobile crowdsensing paradigm called semi-opportunistic mobile crowdsensing. The motivation behind our work in this paper is as follows. For traditional opportunistic mobile crowdsensing, each user moves to predetermined destination along a fixed path. However, in reality, a user may take one path from multiple choices (e.g., most frequently taken paths or most economical paths) to reach the same destination without much disturbance of her daily routine. For a mobile user, each such path is a reasonably good choice for her to make the trip. For the platform, the task assignment performance can be significantly improved since tasks are typically randomly distributed at different locations and in this case the platform has much more choices for task assignment due to the increased path set from the user side. In this way, task coverage probability can be greatly improved with slightly increased human involvement.

In this paper, we first propose the semi-opportunistic mobile crowdsensing paradigm. In this paradigm, each user provides multiple candidate paths for the service platform to choose from. We describe in details how this new paradigm works. We then formulate the optimal task assignment problem, which jointly

chooses mobile users and their taken paths and also tasks for them to perform with an objective to maximize the total task quality subject to given incentive budget constraint and user travel time constraints. We prove this problem is NP-hard and then propose two efficient heuristic algorithms to address this problem. First, we propose a Best Path/Task first algorithm (BPT) which always chooses current best path and current best task into the assignment list. Second, we propose an LP-relaxation based algorithm (LPR), which greedily assigns paths and tasks with the largest values in LP relaxation solution. We deduce the computational complexities of the proposed algorithms. We evaluate the performance of our algorithms on real-world traces. Simulation results show our proposed algorithms are very close to the optimal solution and significantly outperform the case when pre-determined single path is used for each user.

The rest of this paper is organized as follows. We describe the system model and formulate the problem under study in Sect. 2. We propose our algorithms in Sect. 3. We present the simulation results in Sect. 4. We conclude the paper in Sect. 5.

2 Problem Statement

In this section, we describe the semi-opportunistic sensing system, formulate the problem under study, and prove its NP hardness.

2.1 Crowdsensing Paradigm

In this subsection, we introduce the semi-opportunistic sensing system, which consists of a crowdsensing service platform and a set of mobile users.

Mobile User. A mobile user is going to some place in a future time interval (e.g., 7:00–9:00 tomorrow morning). She has a travel time budget (e.g., two hours of commuting time), and fortunately, the actual travel time is less than that. She can use the remaining time to perform some sensing tasks on her travel path to make some money. To reach her destination, she may have multiple candidate paths for the trip. Due to different traffic conditions, each path has distinct travel time and thus distinct remaining time for task performing. She can ask the service platform for compensation of her task performing behavior. She then submits her application form containing her suggested paths to the service platform along with corresponding estimated available task performing time and also demanded compensation for each path. Without causing confusion, hereafter, we shall use the terms “user” and “worker” interchangeably unless otherwise stated.

Service Platform. The crowdsensing service platform receives a set of task performing requests which need to be done in a specific future time interval with a total incentive budget to support the task performing. Each task is submitted with detailed task attributes including task location, task performing time, and task quality requirement. We will discuss the task model in Subsect. 2.2. The platform also receives a list of task performing applications with detailed path

information from mobile users who are willing to perform tasks in this time interval. The platform jointly chooses mobile users and their taken paths and also tasks for them to perform. The objective is to maximize the sum of task quality subject to total incentive budget and user’s path travel time budget.

Semi-opportunistic Crowdsensing Paradigm. Our crowdsensing paradigm is composed of the following steps:

- First, the crowdsensing service platform announces a specific future time interval and recruits mobile user to perform tasks in this time interval.
- Second, each mobile user who is willing to undertake tasks in that future time interval submits to the service platform her suggested moving paths in that time interval, with corresponding task performing time and demanded compensation for each path.
- Third, the platform selects mobile users to perform tasks and informs each selected user which path to take and also which task(s) to perform.
- Fourth, each mobile user travels along the designated path, performs assigned task(s), and finally receives demanded compensation.

2.2 System Model

The three layer task assignment process is illustrated in Fig. 1. The platform selects a set of mobile users to perform tasks on their paths. For each user, one path in her given path set is selected by the platform, if the user is chosen for performing task(s). Some task(s) on this path are then assigned to the user as long as her travel time budget is not violated.

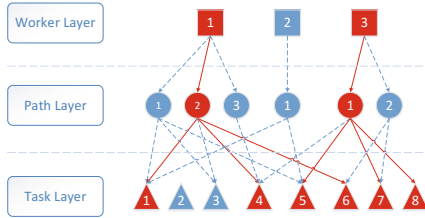


Fig. 1. Three layer task assignment process. The red ones are selected. (Color figure online)

Worker Model. We denote the users by user set $\mathcal{I} = \{1, 2, \dots, I\}$. Each user has a starting point and a destination point. Each user has several path choices for her to reach her destination. We define decision variable $x_i = 1$ if the user $i \in \mathcal{I}$ is selected to perform task, or otherwise $x_i = 0$.

Path Model. Each user has multiple path choices to reach her destination. We denote the path set of user $i \in \mathcal{I}$ as path set $\mathcal{K}_i = \{1, 2, \dots, K_i\}$. Each path

is a simple path from a departure place to a destination passing through a few streets or blocks. Each path $k_i \in \mathcal{K}_i$ has a fixed incentive reward r_{ik_i} and a time budget T_{ik_i} for performing tasks on this path. Different workers demand different incentive rewards for different paths which depend on time budgets of the paths and labor costs per unit time for recruiting the users. We define decision variable $y_{ik_i} = 1$ if user i is selected to travel along path k_i , or otherwise $y_{ik_i} = 0$. Each user can only take one of her provided path choices, and thus we have:

$$x_i = \sum_{k_i} y_{ik_i}, \quad \forall i \in \mathcal{I} \quad (1)$$

The total incentive cannot exceed the platform given incentive budget B , thus we have:

$$\sum_i \sum_{k_i} r_{ik_i} \cdot y_{ik_i} \leq B \quad (2)$$

Task Model. We denote the task set as $\mathcal{J} = \{1, 2, \dots, J\}$. Each task $j \in \mathcal{J}$ is associated with a specific location, task performing time t_j , and task quality q_j . Due to different task types and performing difficulties, tasks have different performing times. Task quality can be indicated by the information value of the task location [11] or the probability of giving a correct answer for an event by majority voting [5]. We define $w_{ik_{ij}} = 1$ if task $j \in \mathcal{J}$ is along the path k_i of user i , or otherwise $w_{ik_{ij}} = 0$. We define decision variable $z_{ik_{ij}} = 1$ if task $j \in \mathcal{J}$ is assigned to user i on her path $k_i \in \mathcal{K}_i$, or otherwise 0. We have

$$z_{ik_{ij}} \leq w_{ik_{ij}}, \quad \forall j \in \mathcal{J}, i \in \mathcal{I}, k_i \in \mathcal{K}_i \quad (3)$$

Each path is associated with zero, one, or several tasks to perform and the total task performing time cannot exceed the path-related task performing time budget. Therefore, we have:

$$\sum_j t_j \cdot z_{ik_{ij}} \leq T_{ik_i} \cdot y_{ik_i}, \quad \forall i \in \mathcal{I}, k_i \in \mathcal{K}_i \quad (4)$$

Task j can only be performed by user i taking path k_i if path k_i of user i is selected by the platform, thus we have:

$$z_{ik_{ij}} \leq y_{ik_i}, \quad \forall j \in \mathcal{J}, i \in \mathcal{I}, k_i \in \mathcal{K}_i \quad (5)$$

We assume each task needs to be done no more than once. Thus, we have

$$\sum_i \sum_{k_i} z_{ik_{ij}} \leq 1, \quad \forall j \in \mathcal{J} \quad (6)$$

2.3 Problem Formulation

The task assignment problem of maximizing the sum of quality of all the tasks subject to the incentive budget constraint and travel time constraints is formulated as follows:

$$\max \sum_{i, k_i, j} q_j \cdot z_{ik_{ij}} \quad (7)$$

$$\begin{aligned}
& s.t. \quad (1) - (6) \\
& var. \quad x_i \in \{0, 1\}, \quad \forall i \in \mathcal{I} \\
& \quad y_{ik_i} \in \{0, 1\}, \quad \forall k_i \in \mathcal{K}_i, \forall i \in \mathcal{I} \\
& \quad z_{ik_ij} \in \{0, 1\}, \quad \forall k_i \in \mathcal{K}_i, \forall i \in \mathcal{I}, \forall j \in \mathcal{J}
\end{aligned}$$

The hardness of the above problem is given as follows. We consider a simplified case where there is only one user who has exactly one path to take and the platform can afford her incentive reward. Then the problem is to maximize the total task quality under the user’s task performing time budget. This is a 0-1 knapsack problem. Here, the item weight is task performing time and the sum of item weights cannot exceed the weight capacity (i.e., the user’s task performing time budget). Moreover, here, the value of an item is the quality of a task and the objective is to maximize the sum of values of chosen items. The 0-1 knapsack problem is known to be NP-hard. Therefore, the problem under study in this paper is also an NP-hard problem.

3 Proposed Task Assignment Algorithms

There are two resource constraints (i.e., incentive reward and per-path-related task performing time) in the task assignment problem under study. Incentive reward constraint is a high-layer constraint from the service platform and path-related task performing time constraint is a low-layer constraint from the user side. In this section, we propose two efficient heuristic algorithms to address the NP-hard task assignment problem formulated in Subsect. 2.3. First, we propose Best Path/Task First Algorithm (BPT) which always chooses current best path and current best task into the assignment list. Second, we propose an LP-relaxation based algorithm, which greedily assigns paths and tasks with largest values into the LP relaxation solution.

3.1 Best Path/Task First Algorithm (BPT)

In this subsection, we propose BPT algorithm. The algorithm is composed of two major components: TASK SELECTION and PATH SELECTION.

TASK SELECTION. For each of the candidate paths provided by users, the TASK SELECTION process selects a set of still available tasks leading to the maximum task quality while meeting the path-related task performing time constraint. This is a 0-1 knapsack problem and is known to be NP-hard. Here, an approximate algorithm is presented for the TASK SELECTION, which works as follows.

1. Sort all still available tasks on this path in the descending order according to the ratio of task quality and performing time.
2. If the sum of task performing time of all still available tasks is no larger than task performing time budget, then all tasks are selected.

3. Otherwise, find the maximum number m such that the first m tasks in the task list meet the following condition: These m tasks' total task performing time is smaller than or equal to the task performing time budget while the first $(m + 1)$ tasks' total task performing time is larger than the task performing time budget. If the sum of task quality of the first m tasks are smaller than the task quality of the $(m + 1)^{th}$ task and the task performing time of the $(m + 1)^{th}$ task is not larger than the task performing time budget, then the $(m + 1)^{th}$ task is selected, otherwise the first m tasks are selected.

The above steps for TASK SELECTION work in spirit similar to the 2-approximate algorithm in [1] and thus also has an approximate ratio of two.

PATH SELECTION. This process aims to select a set of users and designate a specific path to each of them with an objective of maximizing the total task quality under incentive budget. For this purpose, we propose a greedy algorithm which works as follows.

1. Compute the sum of task quality of tasks available to be performed on each candidate path using the above TASK SELECTION algorithm.
2. Compute the quality/incentive ratio (i.e., the ratio of sum of quality of all tasks (temporarily) assigned to a path to the path-related incentive reward) of each path.
3. Select the path with the largest ratio and go to step 1 if the incentive budget constraint is not violated after the selection.
4. If the incentive budget constraint is violated after adding the path in the above step, and if the total task quality of this path is larger than the sum of total task quality of all previously selected paths, then abandon all the previously-selected paths and take this path.

The detailed procedure for BPT is given in Algorithm 1. Lines 1–2 are for initialization. Lines 3–40 iteratively choose the path leading to the largest quality/incentive ratio while Lines 10–26 give detailed procedure for selecting tasks for each candidate path. More specifically, line 3 ensures that the incentive budget constraint is not violated. In line 4, i^* , k^* , and Q^* record the user, path, and path-specific total task quality for the so far best path, respectively. P records the ratio of total task quality and incentive for the so far best path, whose initial value is zero. Lines 6–8 ensure that at most one path is selected for each user. Line 10 ensures to choose tasks which are on the current path and further have not been selected by any other path. Lines 12–13 consider the case when the task performing time budget is large enough to cover all the tasks on the path. Lines 14–22 determine whether to take the first m tasks or the $(m + 1)^{th}$ task. Lines 24–26 select the path with the largest ratio of overall task quality to the incentive for taking the path. Lines 30–32 add paths into the assignment list if the incentive budget constraint is not violated. Lines 34–38 take the last path and abandon all previously-selected paths if the task quality of the last path is larger than the sum of all the previously-selected paths.

The computational complexity for the TASK SELECTION in lines 10–26 is $O(J \log(J))$ due to the task sorting operation. Thus, the computational com-

plexity of BPT is $O(I^2 K J \log(J))$, where I , K , J denote the number of users, number of path choices of a user, and number of tasks, respectively.

Algorithm 1 Procedure for BPT.

Input: $\mathcal{I}, \mathcal{K}_i, \mathcal{J}, B, r_{ik_i}, T_{ik_i}, t_j, q_j, w_{ik_i j}$
Output: $x_i, y_{ik_i}, z_{ik_i j}$

```

1: set any  $x_i, y_{ik_i}, z_{ik_i j}$  to 0.
2:  $sumB \leftarrow 0, Q^+ \leftarrow 0$ 
3: while  $sumB < B$  do
4:    $i^* \leftarrow null, k^* \leftarrow null, Q^* \leftarrow 0, P \leftarrow 0$ 
5:   for  $i \in \mathcal{I}$  do
6:     if  $x_i = 1$  then
7:       continue;
8:     end if
9:     for  $k_i \in \mathcal{K}_i$  do
10:       $\mathcal{J}_{ik_i} \leftarrow \left\{ j \mid \forall j, w_{ik_i j} = 1, \sum_i \sum_{k_i} z_{ik_i j} = 0 \right\}$ 
11:      Sort  $j \in \mathcal{J}_{ik_i}$  in the descending order of  $q_j/t_j$ 
12:      if  $\sum_{j \in \mathcal{J}_{ik_i}} t_j \leq T_{ik_i}$  then
13:         $Q_{ik_i} \leftarrow \sum_{j \in \mathcal{J}_{ik_i}} q_j$ 
14:      else
15:         $m^* \leftarrow \arg \max_m \sum_{j \in \text{the first } m \text{ tasks in } \mathcal{J}_{ik_i}} t_j \leq T_{ik_i} <$ 
 $\sum_{j \in \text{the first } m+1 \text{ tasks in } \mathcal{J}_{ik_i}} t_j$ 
16:         $j^* \leftarrow \text{the } (m^* + 1)^{th} \text{ task in } \mathcal{J}_{ik_i}$ 
17:        if  $\sum_{j \in \text{the first } m^* \text{ tasks in } \mathcal{J}_{ik_i}} q_j < q_{j^*}$  and  $t_{j^*} \leq T_{ik_i}$  then
18:           $Q_{ik_i} \leftarrow q_{j^*}, \mathcal{J}^* \leftarrow \{j^*\}$ 
19:        else
20:           $Q_{ik_i} \leftarrow \sum_{j \in \text{the first } m^* \text{ tasks in } \mathcal{J}_{ik_i}} q_j$ 
21:           $\mathcal{J}^* \leftarrow \{\text{the first } m^* \text{ tasks in } \mathcal{J}_{ik_i}\}$ 
22:        end if
23:      end if
24:      if  $Q_{ik_i}/r_{k_i} > P$  then
25:         $P \leftarrow Q_{ik_i}/r_{k_i}, Q^* \leftarrow Q_{ik_i}, i^* \leftarrow i, k^* \leftarrow k_i$ 
26:      end if
27:    end for
28:  end for
29:   $sumB \leftarrow sumB + r_{i^* k^*}$ 
30:  if  $sumB \leq B$  then
31:     $x_{i^*} \leftarrow 1, y_{i^* k^*} \leftarrow 1, Q^+ \leftarrow Q^+ + Q^*$ 
32:     $z_{i^* k^* j} \leftarrow 1, \forall j \in \mathcal{J}^*$ 
33:  else
34:    if  $Q^* > Q^+$  and  $r_{i^* k^*} \leq B$  then
35:      set any  $x_i, y_{ik_i}, z_{ik_i j}$  to 0.
36:       $x_{i^*} \leftarrow 1, y_{i^* k^*} \leftarrow 1$ 
37:       $z_{i^* k^* j} \leftarrow 1, \forall j \in \mathcal{J}^*$ 
38:    end if
39:  end if
40: end while

```

3.2 LP-Relaxation Based Algorithm(LPR)

We propose an LP-relaxation based algorithm, which greedily assigns paths and tasks with largest values in Linear Programming(LP) relaxation solution. The algorithm is composed of LP-relaxation based initialization and greedy task assignment.

LP-Relaxation Based Initialization. The 0–1 integer constraints of x_i , y_{ik_i} , and z_{ik_ij} are relaxed as follows:

$$0 \leq x_i \leq 1, 0 \leq y_{ik_i} \leq 1, 0 \leq z_{ik_ij} \leq 1, \quad \forall j \in \mathcal{J}, i \in \mathcal{I}, k_i \in \mathcal{K}_i \quad (8)$$

Algorithm 2 Procedure for LPR.

Input: $\mathcal{I}, \mathcal{K}_i, \mathcal{J}, B, r_{ik_i}, T_{ik_i}, t_j, q_j, w_{ik_ij}$

Output: x_i, y_{ik_i}, z_{ik_ij}

```

1: set any  $x_i, y_{ik_i}, z_{ik_ij}$  to 0.
2:  $x_i^*, y_{ik_i}^*, z_{ik_ij}^* \leftarrow$  Run LP algorithm
3:  $sumB \leftarrow 0, Q^+ \leftarrow 0$ 
4: while  $sumB < B$  do
5:    $y^* \leftarrow$  largest  $y_{ik_i}^*$  (if user  $i$  has not been selected yet)
6:    $i^* \leftarrow arg_i$  largest  $y_{ik_i}^*, k^* \leftarrow arg_{k_i}$  largest  $y_{ik_i}^*$ 
7:    $sumB \leftarrow sumB + r_{i^*k^*}$ 
8:   if  $sumB \leq B$  then
9:      $x_{i^*} \leftarrow 1, y_{i^*k^*} \leftarrow 1$ 
10:     $sumT \leftarrow 0$ 
11:    while  $sumT < T_{i^*k^*}$  do
12:       $z^* \leftarrow$  largest  $z_{i^*k^*j}^*$  (if task  $j$  has not been selected yet)
13:       $j^* \leftarrow arg_j$  largest  $z_{i^*k^*j}^*$ 
14:       $sumT = sumT + t_{j^*}$ 
15:      if  $sumT \leq T_{i^*k^*}$  then
16:         $z_{i^*k^*j^*} \leftarrow 1$ 
17:      end if
18:    end while
19:  end if
20: end while

```

After the relaxation, the new problem can be solved by a linear programming solver in polynomial time. And we use $x_i^*, y_{ik_i}^*, z_{ik_ij}^*$ to denote the optimal solution of this linear programming problem.

Greedy Task Assignment. The greedy task assignment is composed of path selection and task selection. The algorithm iteratively selects the path with the highest value of $y_{ik_i}^*$ if the user of this path has not yet been selected and the total incentive budget is not violated after selecting this path. For each selected path, the algorithm iteratively selects the task with the highest value of $z_{ik_ij}^*$ if the task j has not yet been selected and the total task performing time budget of this path is not violated after selecting this task. This algorithm is feasible since the incentive budget constraint and task performing time constraint are not violated.

LPR is described in Algorithm 2. Line 2 runs the linear programming algorithm to obtain the relaxed solution. Lines 5–9 select the path with the maximum

value of $y_{ik_i}^*$ into the assignment list. Lines 11–18 iteratively select the task with the maximum value of $z_{i^*k^*j}^*$ into the assignment list.

The computational complexity of solving linear programming problem is polynomial. The computational complexity of Greedy Task Assignment is $O(IKJ)$. Therefore the computational complexity of LPR is polynomial.

4 Performance Evaluation

In this section, we conduct extensive simulations to evaluate the performance of our proposed algorithms. We shall describe detailed simulation settings, the traces that we use, and finally present the simulation results.

We compare our algorithms with optimal solution on real-world traces. The optimal solution is obtained by using branch and cut algorithm which is a time-consuming algorithm. We denote the optimal solution as *Semi_opt*. We also compare our algorithms with the traditional opportunistic sensing paradigm where each mobile user only takes one pre-determined path which is her most frequently taken path. The optimal solution for this traditional opportunistic sensing paradigm is referred to as *Oppt_opt*. In addition, in our simulations, average task quality equals to total task quality divided by total task number. The default parameter settings are shown in Table 1.

Table 1. Default parameter settings.

Parameters	Values
Path number for a user	$N(3, 1)^1$
Worker number	20
Task number	200
Task performing time of a task	$N(20, 10)$
Task performing time budget for a path	$N(100, 30)$
Path incentive	$N(100, 30)$
Task quality	$N(100, 30)$

¹ $N(x, y)$ means normal distribution whose mean value is x and standard deviation is y

We use two real-world traces Geolife dataset [13] and NCCU dataset [6] to evaluate the performance of our proposed algorithms. Geolife dataset [13] was collected by 182 users and contains 17,621 trajectories. The majority of the data was created in Beijing, China. We use trajectories in main urban zone of Beijing city which is of $17\text{ km} \times 18\text{ km}$. The transportation mode of these users includes driving, taking a bus, riding a bike, and walking. NCCU dataset [6] was collected by 115 college students for 15 days in a campus environment of $3764\text{ m} \times 3420\text{ m}$. Similar to [10], we divide the areas into square subareas. Mobile users move

among different subareas and user paths are sequences of subarea indexes. We divide the area of Geolife dataset into 20×20 subareas and divide the area of NCCU dataset into 10×10 subareas. In these two datasets, paths were chosen based on frequencies that they were taken in the datasets.

We evaluate the average performance of different algorithms on real-world traces. In Figs. 2(a) and 2(b), as the number of paths K increases, average task quality of *Semi_opt*, LPR, and BPT increases as well. The performance of LPR and BPT is close to *Semi_opt*. The performance of *Semi_opt* and that of *Oppt_opt* are the same when $K = 1$ since they both choose the most frequently taken path for each user and use optimal algorithm for task assignment. When $K > 1$, *Semi_opt*, LPR, and BPT have better performance than *Oppt_opt* which again validates the effectiveness of our proposed crowdsensing paradigm.

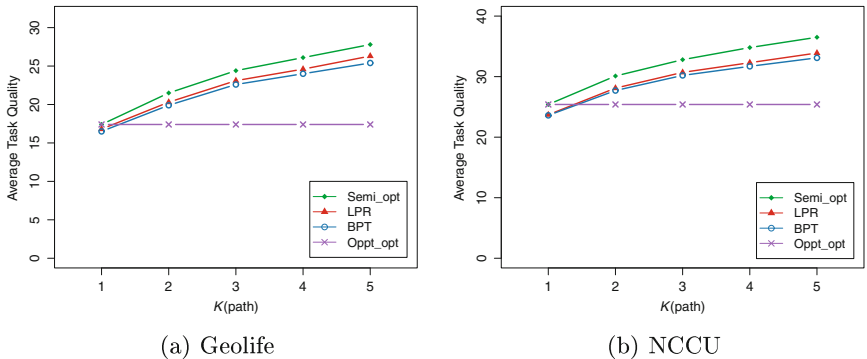


Fig. 2. Performance comparison of different algorithms on real-world traces.

5 Conclusion

In this paper, we proposed a new mobile crowdsensing paradigm called semi-opportunistic sensing where each mobile user has multiple path choices to take to reach her destination. We formulated the optimal task assignment problem, which aims to select proper mobile users taking proper paths and assign tasks to maximize the total task quality while meeting given incentive budget constraint and travel time constraints. We proved the problem is NP-hard and proposed two efficient algorithms. We presented detailed algorithm design and deduced their computational complexities. We conducted extensive simulations on real-world traces. Simulation results validate the effectiveness of our proposed crowdsensing paradigm and further the results show that the performance of our proposed algorithms is close to the optimal solution.

References

1. Du, D.Z., Ko, K.I., Hu, X.: Design and Analysis of Approximation Algorithms. Springer, New York (2012). <https://doi.org/10.1007/978-1-4614-1701-9>
2. Ganti, R.K., Ye, F., Lei, H.: Mobile crowdsensing: current state and future challenges. *IEEE Commun. Mag.* **49**(11), 32–39 (2011)
3. Gao, R., et al.: Jigsaw: indoor floor plan reconstruction via mobile crowdsensing. In: Proceedings of ACM MobiCom 2014, pp. 249–260 (2014)
4. Gong, W., Zhang, B., Li, C.: Task assignment in mobile crowdsensing: present and future directions. *IEEE Netw.* <https://doi.org/10.1109/MNET.2018.1700331>
5. Gong, W., Zhang, B., Li, C.: Location-based online task scheduling in mobile crowdsensing. In: Proceedings of IEEE GLOBECOM 2017, Singapore, pp. 1–6, December 2017
6. Tsai, T.C., Chan, H.H.: NCCU trace: social-network-aware mobility trace. *IEEE Commun. Mag.* **53**(10), 144–149 (2015)
7. Wang, X., Zhang, J., Tian, X., Gan, X., Guan, Y., Wang, X.: Crowdsensing-based consensus incident report for road traffic acquisition. *IEEE Trans. Intell. Transport. Syst.*, October 2017. <https://doi.org/10.1109/TITS.2017.2750169>
8. Xu, C., Li, S., Zhang, Y., Miluzzo, E., Chen, Y.: Crowdsensing the speaker count in the wild: implications and applications. *IEEE Commun. Mag.* **52**(10), 92–99 (2014)
9. Yang, F., Lu, J.L., Zhu, Y., Peng, J., Shu, W., Wu, M.Y.: Heterogeneous task allocation in participatory sensing. In: Proceedings of IEEE GLOBECOM 2015, pp. 1–6, December 2015
10. Zhang, B., Song, Z., Liu, C.H., Ma, J., Wang, W.: An event-driven qoi-aware participatory sensing framework with energy and budget constraints. *ACM Trans. Intell. Syst. Technol.* **6**(3), 1–19 (2015)
11. Zhang, M., et al.: Quality-aware sensing coverage in budget-constrained mobile crowdsensing networks. *IEEE Trans. Veh. Technol.* **65**(9), 7698–7707 (2016)
12. Zheng, Y., Liu, F., Hsieh, H.: U-Air: when urban air quality inference meets big data. In: Proceedings of ACM KDD 2013, pp. 1436–1444, August 2013
13. Zheng, Y., Xie, X., Ma, W.Y.: GeoLife: a collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.* **33**(2), 32–40 (2010)