Toward Optimal Allocation of Location Dependent Tasks in Crowdsensing

Shibo He[†], Dong-Hoon Shin[†], Junshan Zhang[†], Jiming Chen[‡]
[†]School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ, 85287, USA
[‡]State Key Laboratory of Industrial Control Technology, Zhejiang University, China
Email: {Shibo.He, Donghoon.Shin.2, Junshan.Zhang}@asu.edu, jmchen@iipc.zju.edu.cn

***Abstract*—*Crowdsensing* offers an efficient approach to meet the demand in large scale sensing applications. In *crowdsensing*, it is of great interest to find the optimal task allocation, which is challenging since sensing tasks with different requirements of quality of sensing are typically associated with specific locations and mobile users are constrained by time budgets. We show that the allocation problem is NP hard. We then focus on approximation algorithms, and devise an efficient local ratio based algorithm (LRBA). Our analysis shows that the approximation ratio of the aggregate rewards obtained by the optimal allocation to those by LRBA is 5. This reveals that LRBA is efficient, since a lower (but not tight) bound on the approximation ratio is 4. We also discuss about how to decide the fair prices of sensing tasks to provide incentives since mobile users tend to decline the tasks with low incentives. We design a pricing mechanism based on bargaining theory, in which the price of each task is determined by the performing cost and market demand (i.e., the number of mobile users who intend to perform the task). Extensive simulation results are provided to demonstrate the advantages of our proposed scheme.**

***Index Terms*—Crowdsensing Applications, Location Dependent Task Allocation, Approximation Ratio**

I. INTRODUCTION

Thanks to the technological advances, small-sized portable mobile devices are becoming extremely prevailing nowadays. Well-known examples include tablet computer (e.g., iPad), smartphone (e.g., Google Nexus), etc. These pocket-sized mobile devices are also typically integrated with a set of sensors (e.g., camera, light sensor, chemical sensor and GPS). Other types of sensors, such as health measuring and environment monitoring sensors, are currently being developed, and are expected to be available in mobile devices in the near future [1]. Such diverse sensors provide a high volume of data about the environment, human society and individuals. This, along with the dramatic proliferation of mobile devices, accelerates the emergence of *crowdsensing* applications [2]–[5].

Roughly speaking, *crowdsensing* refers to leveraging the power of large crowd to complete the tasks of large scale sensing applications at a lower cost. This, however, calls for an interface system which allows mobile users to participate in the applications and to perform the sensing tasks. There has been much attention paid to build up such an interface system that enables a specific *crowdsensing* application. Dutta *et al.* [6], for example, proposed a system, named *Common Sense*, to measure the air quality of a city. In this, people move around the city carrying mobile devices that are integrated with a set of chemical sensors, such as carbon monoxide. The data about air quality levels and their associated locations are measured during the movement, and are uploaded to a backend cloud. They are then fused and visualized. *Common Sense* provides a dynamic and fine-grained air quality level of the whole city. This, without the involvement of the crowd, would be costly and even be impossible to achieve. Other similar systems have been proposed to make *crowdsensing* possible for other environmental, infrastructural and social applications [7]–[12].

While many existing works have demonstrated the advantages of *crowdsensing* for a specific sensing application, it would be more interesting and economical to build a unified platform for a variety of applications, considering that systems for many sensing applications bear strong similarity. Notably, Ra *et al.* [13] proposed a programable unified platform, referred to as *Medusa*. A prominent characteristic of *Medusa* is that it provides an incentive-based mechanism, where monetary incentives are offered to guarantee the quality of sensing. With an incentive-based unified platform, anyone who has sensing application demand can publish it in the platform, and also anyone who wants to participate for incentive reward can claim a list of interested sensing tasks.

Clearly, the unified platform brings down the implementation cost, and enables easy management. Nevertheless, it also engenders some challenging issues. One of them is the efficient allocation of sensing tasks, which is fundamentally different from general allocation problems due to the unique requirements of sensing tasks and mobile users. To elaborate, in most *crowdsensing* applications, each sensing task has to be performed at a specific location, e.g., measure the air quality at a given location. Therefore, it takes a certain amount of time for a mobile user to travel around in order to perform the allocated tasks. Since each mobile user has a limited time budget each day, he or she has limited capacity in performing the sensing tasks. Moreover, in general it is difficult to know how sincerely each mobile user would perform the tasks, and thus it is difficult to guarantee the quality of sensing.

In this paper, we take the first attempt to explore the task allocation problem by taking into account the above two unique requirements that bring up new challenges. First, the total traveling time to perform a set of sensing tasks depends on both the locations of mobile users and the locations of sensing tasks. A small change in the allocated set of

This research was supported in part by the U.S. National Science Foundation under Grants CNS1117462 and CNS-1218484, NSFC under grants 61222305, SRFDP under grant 20120101110139, NCET-11-0445 and National Program for Special Support of Top-Notch Young Professionals.

sensing tasks may cause a prominent change in total traveling time, making it extremely difficult to satisfy the time budget constraint for each mobile user. Second, multiple mobile users may need to be coordinated to perform the same sensing task, so as to provide the required quality of sensing. Further, complementary to the task allocation, we discuss on how to design a fair and efficient pricing mechanism for both mobile users and the platform to reach an agreement on the price of each sensing task. Our contributions are three folds.

- We study the problem of allocating location dependent tasks, by taking into account both the geographical characteristics of sensing tasks and the spatial movement constraints of mobile users. We mathematically formulate this problem, and show that the formulated problem is NP hard.
- We design an efficient approximation algorithm, namely local ratio based algorithm (LRBA), to solve the proposed allocation problem. LRBA decomposes the problem into several subproblems by modifying the reward function at each iteration, and it is thus easy to implement. We analyze the performance of LRBA, i.e., the approximate ratio of the aggregate rewards obtained by the optimal to those by LRBA, and show that LRBA is a $5-$approximate algorithm. Note that a lower but not tight bound on the approximate ratio is 4. Hence, LRBA achieves reasonable worse-case performance.
- We design a pricing mechanism based on bargaining theory, for both the platform and mobile users to reach an agreement on the prices of sensing tasks. The price of each task is determined by the performing cost and market demand (i.e., the number of mobile users who intend to perform the task). The lower the performing cost or the higher the market demand, the lower the price, which is consistent with the realistic market.

The rest of the paper is organized as follows. We present the system model in Sec. II. We devise an efficient algorithm LRBA in Sec. III to solve the task allocation problem and theoretically investigate the worse-case performance of LRBA in Sec. IV. We discuss how to design an efficient pricing mechanism in Sec. V. We provide simulation results in Sec. VI to demonstrate the performance of LRBA and discuss the related work on *crowdsensing* in Sec. VII. We conclude our work in Sec. VIII.

II. Problem formulation

We consider a crowdsensing platform, via which different sensing tasks can be performed by leveraging the power of the crowd [1]. Due to economic or geographical reasons, individuals with sensing tasks can publish their tasks in the platform and offer some incentives for those who complete these tasks. Available mobile users can register and claim for a list of interested tasks to perform. We denote the collection of registered mobile users by $\mathcal{U} = \{u_1, u_1, \cdots, u_n\}$, and the set of published sensing tasks by $\mathcal{T} = \{t_1, t_2, \cdots, t_m\}$.

Typically, sensing tasks are location dependent, i.e., each sensing task t_j is associated with a specific location P_{t_j}. For example, sensing data about air quality level or traffic condition have to be measured at some selected locations. We include the geographical characteristic of sensing tasks, since mobile users have to travel to the locations associated with the tasks. Denote by P_{u_i} the original location of each user u_i at the beginning of task performing. Let $\mathcal{T}_i$ denote the set of tasks that the platform assigns to user u_i. To complete multiple tasks, each user has to travel from its current location P_{u_i} to visit each P_{t_j}, $t_j \in \mathcal{T}_i$. Denote by $\mathcal{P}(\mathcal{T}_i)$ the shortest path that starts at P_{u_i} and connects all points in $\mathcal{T}_i$. The total time that each user u_i spends for the sensing tasks $\mathcal{T}_i$ is dominated by the total length of $\mathcal{P}_{u_i}$, denoted by $D(\mathcal{P}(\mathcal{T}_i))$. It is reasonable to assume that each user u_i has a time budget for performing tasks, which corresponds to a traveling distance budget B_i.

To guarantee the quality of sensing of each task t_j, sensing redundancy is necessary in many applications, e.g., multiple measurements of air quality can be fused to reduce the measurement error. Assume that each task t_j needs l_j independent sensing measurements to guarantee quality of sensing, where l_j depends on the sensing requirement of task t_j. For simplicity, sensing measurements for the same task from different mobile users can be viewed as a set of new sensing tasks associated with the same locations. The only difference is that each mobile user can perform at most one task from the set.

Let $R_{ij} = r'_{ij} - r_{ij}$ denote the net reward when user u_i performs task t_j, where r'_{ij} is the reward that the platform will receive and r_{ij} is the price that the platform has to pay to user u_i [2]. For convenience, we introduce the decision variables x_{ij}, $u_i \in \mathcal{U}$, $t_j \in \mathcal{T}$: $x_{ij} = 1$ if the platform assigns task t_j to user u_i; $x_{ij} = 0$ otherwise. Then $\mathcal{T}_i$ can be obtained as $\mathcal{T}_i = \{t_j | x_{ij} = 1, v_j \in \mathcal{T}\}$. The problem of maximizing rewards of the platform (MRP) can be formulated as follows:

$$MRP : \max\ g(\mathbf{x}) = \sum_{u_i \in \mathcal{U}} \sum_{t_j \in \mathcal{T}} R_{ij} x_{ij}$$

$$s.t. \begin{cases} \sum\limits_{u_i \in \mathcal{U}} x_{ij} \leq l_j,\ t_j \in \mathcal{T}. \\ D(\mathcal{P}(\mathcal{T}_i)) \leq B_i,\ u_i \in \mathcal{U}. \\ x_{ij} = \{0, 1\},\ u_i \in \mathcal{U}, t_j \in \mathcal{T}. \end{cases}$$

In MRP, the second inequality is used to ensure that each mobile user's traveling distance budget is not violated. In the first inequality, we adopt $\sum\limits_{u_i \in \mathcal{U}} x_{ij} \leq l_j$ instead of $\sum\limits_{u_i \in \mathcal{U}} x_{ij} = l_j$ since we consider a maximization problem and there are large number of mobile users in the system.

III. Allocating Sensing Tasks

In this section, we will first discuss about the hardness of MRP and introduce the notion of Local Ratio [14], [15]. Then we elaborate on the local ratio based algorithm (LRBA) design and give an example to show how LRBA works. The theoretical analysis about the performance of LRBA is performed in the next section.

[1]Please refer to [13] for the implementation of such a platform.

[2]We will discuss how to decide price r_{ij} in Sec. V.

A. The hardness of MRP

In this subsection, we will show that MRP is an NP hard problem. To this end, we try to degenerate MRP to a special case where there is only one mobile user u_i. In this case, we can model the system as a complete graph $G(V_i, E_i)$, where $V_i = u_i \cup \mathcal{T}$ and E_i denotes the edges between any two vertices in V_i. The weight of each edge $e_{kk'}$ represents the distance between vertices k and k' (k and k' can be either u_i or a point from $\mathcal{T}$). For each vertex k satisfying $k \in \mathcal{T}$, its reward is R_{ik}, and the reward of vertex u_i equals to 0. The goal is to find a path $\mathcal{P}$ originated at u_i with total length no more than B_i, such that the total rewards gained from the vertices in the path $\mathcal{P}$ is maximized. According to [14], we know the special case of MRP is an *orienteering* problem, which is known to be NP hard. We have the following result.

Theorem 1: MRP is NP hard.

Proof: Let $|\mathcal{U}| = 1$. As aforementioned, MRP is degenerated to an *orienteering* problem. If MRP can be solved in polynomial time, then so does *orienteering* problem, which is a contradiction. ■

Since MRP is NP hard, we focus on approximation algorithms. We note that MRP is not a *team orienteering* problem though its special case for each mobile user can be reduced to an *orienteering* problem. There are mainly two differences: 1) in a *team orienteering* problem, the reward gained at each vertex k is the same, while in MRP, different mobile users can gain different rewards for different vertices; 2) in a *team orienteering* problem, each vertex can only be visited once while in MRP multiple visits are allowed. Therefore, MRP can not be solved by existing algorithms designed for *team orienteering* problem.

In the rest of this work, we focus on a constrained integer maximization problem, which comprises of an objective function $\varphi(\cdot)$ and a feasible set $\mathcal{C}$. We will use $(\mathcal{C}, \varphi(\cdot))$ or $\varphi(\cdot)$ when there is no confusion, to represent a constrained integer maximization problem.

B. Local Ratio and the outline of LRBA

We begin with introducing the definition of γ-approximate algorithm and γ-approximate solution.

Definition 1: An algorithm for a maximization problem $(\mathcal{C}, \varphi(\cdot))$ is a γ-approximate algorithm if the solution $\boldsymbol{x}$ that it obtains satisfies $\gamma\varphi(\boldsymbol{x}) \geq \varphi(\boldsymbol{x}^*)$ for any instance, where $\boldsymbol{x}^*$ is the optimal solution. $\boldsymbol{x}$ is referred to as a γ-approximate solution of $(\mathcal{C}, \varphi(\cdot))$.

The optimal solution of an integer maximization problem depends on the feasible set $\mathcal{C}$ and the objective function $\varphi(\cdot)$. The following local ratio theorem discusses about the relationships among solutions for different objective functions when $\mathcal{C}$ remains the same.

Theorem 2 (Local Ratio [15]): Let $\varphi(\cdot)$, $\varphi_1(\cdot)$, $\varphi_2(\cdot)$, $\cdots$, $\varphi_N(\cdot)$, are $N+1$ objective functions satisfying $\varphi(\cdot) = \varphi_1(\cdot) + \varphi_2(\cdot) + \cdots + \varphi_N(\cdot)$. If $\boldsymbol{x}$ is a γ-approximate solution of $(\mathcal{C}, \varphi_i(\cdot))$, $i = 1, 2, \cdots, N$, then $\boldsymbol{x}$ is also a γ-approximate solution of $(\mathcal{C}, \varphi(\cdot))$.

Local ratio theorem provides an approach to decompose a complex problem $(\mathcal{C}, \varphi(\cdot))$ into several simple subproblems $(\mathcal{C}, \varphi_i(\cdot))$. Note that the reward function of MRP, i.e., $g(\boldsymbol{x}) = \sum_{u_i \in \mathcal{U}} \sum_{t_j \in \mathcal{T}} R_{ij} x_{ij}$, is a linear function of x_{ij}, and can be characterized by the reward R_{ij}, $u_i \in \mathcal{U}, t_j \in \mathcal{T}$. We can also define another reward function by modifying the reward R_{ij}, e.g.,

$$g_1(\boldsymbol{x}) = \sum_{u_i \in \mathcal{U}} \sum_{t_j \in \mathcal{T}} R'_{ij} x_{ij},$$

where $R'_{ij} = R_{ij}$ for $i = 1$ and $R'_{ij} = 0$ for $i \neq 1$. Similarly, let

$$g_2(\boldsymbol{x}) = \sum_{u_i \in \mathcal{U}} \sum_{t_j \in \mathcal{T}} R''_{ij} x_{ij},$$

where $R''_{ij} = R_{ij}$ for $i \neq 1$ and $R''_{ij} = 0$ for $i = 1$. Obviously, we have $g(\boldsymbol{x}) = g_1(\boldsymbol{x}) + g_2(\boldsymbol{x})$. We still denote by $\mathcal{C}$ the feasible set of MRP. Note that $(\mathcal{C}, g_1(\boldsymbol{x}))$ is actually the *orienteering* problem associated with mobile user u_1. Repeating the process iteratively, we can decompose the original MRP problem into n *orienteering* subproblems. The challenge lies in how to decide each $g_i(\boldsymbol{x})$ such that the resultant solution $\boldsymbol{x}$ is an approximate solution to each subproblem $(\mathcal{C}, g_i(\cdot))$. Our LRBA is inspired by existing works [15], [16]. Before we elaborate LRBA in the next subsection, we first give an outline of LRBA in the following.

1) Transforming the original MRP problem. For each sensing task t_j, which requires l_j measurements of sensing data from different users, we regard each copy for a measurement as an independent sensing task associated with the same location P_{t_j} and reward R_{ij}. We thus obtain a new MRP problem $(\mathcal{F}, f(\boldsymbol{y}))$.
2) Solving the *orienteering* problem of each mobile user forwards. For the new MRP problem, from mobile user u_1 to u_n, solve the *orienteering* problem of each user u_i (denote the results by O_i), and then determine the reward function $f_i(\boldsymbol{y})$ iteratively (see Sec. III-C)) such that
$$f(\boldsymbol{y}) = f_1(\boldsymbol{y}) + f_{n-1}(\boldsymbol{y}) + \cdots + f_n(\boldsymbol{y}).$$
3) Refining the final assignment backwards. From user u_n to u_1, assign sensing tasks (denoted by S_i) to each user u_i according to the results O_i. The sensing tasks that have been allocated earlier will be excluded, i.e., $S_i = O_i \setminus \cup_{k=i+1}^{n} S_k$.

C. A detailed description of LRBA

1) Transforming the original MRP problem. In the original MRP problem, each sensing task can be allocated to l_j users to guarantee the quality of sensing, which increases the problem complexity. In LRBA, each sensing task t_j is further divided into l_j independent sensing tasks, all of which are associated with the same location P_{t_j}. We denote the new sensing tasks by $t_{jk}, j = 1, 2, \cdots, m, k = 1, 2, \cdots, l_j$. Hence, we have in total $m' = \sum_{j=1}^{m} l_j$ sensing tasks, and each new sensing task can be allocated to at most one user. The platform can gain

a reward c_{ijk} if user u_i is assigned to perform the sensing task t_{jk}, where $c_{ijk} = R_{ij}$ for all $k = 1, 2, \cdots, l_j$. For easy presentation, we will refer to c_{ijk} as the reward of (i, j, k). To give a formal formulation, we introduce decision variables y_{ijk}: $y_{ijk} = 1$ if the platform assigns task t_{jk} to user u_i, $i = 1, 2, \cdots, n$, $j = 1, 2, \cdots, m$, $k = 1, 2, \cdots, l_j$; $y_{ijk} = 0$ otherwise. Given a feasible solution $\boldsymbol{y}$, let $\mathcal{T}_i' = \{t_{jk} | y_{ijk} = 1, j = 1, 2, \cdots, m, k = 1, 2, \cdots, l_j\}$, i.e, $\mathcal{T}_i'$ is the set of sensing tasks allocated to user u_i. We have the new MRP problem as follows.

$$\text{N-MRP} : f(\boldsymbol{y}) = \max \sum_{u_i \in \mathcal{U}} \sum_{j=1}^{m} \sum_{k=1}^{l_j} c_{ijk} y_{ijk}$$

$$s.t. \begin{cases} \sum\limits_{u_i \in \mathcal{U}} y_{ijk} \le 1,\ j = 1, 2, \cdots, m, k = 1, 2, \cdots, l_j, \\ \sum\limits_{k=1}^{l_j} y_{ijk} \le 1,\ u_i \in \mathcal{U}, j = 1, 2, \cdots, m, \\ D(\mathcal{P}(\mathcal{T}_i')) \le B_i,\ u_i \in \mathcal{U}, \\ y_{ijk} = \{0, 1\}, \end{cases}$$

where the second constraint requires that each user can at most perform one sensing task from set $\{t_{jk} | k = 1, 2, \cdots, l_j\}$. Let $\mathcal{F}$ to denote the feasible solution of N-MRP. The new MRP problem can be expressed as $(\mathcal{F}, f(\boldsymbol{y}))$.

2) Solving the *orienteering* problem of each user forwards. Now we adopt local ratio to solve the new MRP problem iteratively. Since we will modify the reward (objective) function in each iteration, we use subscript $f'_{I-1}(\boldsymbol{y})$ to denote the reward function at the beginning of iteration I, and $f_I(\boldsymbol{y})$ the modified reward function at iteration I. Accordingly, we denote the reward of (i, j, k) associated with $f'_{I-1}(\boldsymbol{y})$ by $c_{ijk}^{(I-1)'}$, and the reward of (i, j, k) associated with $f_I(\boldsymbol{y})$ by $c_{ijk}^{(I)}$ at iteration I.

Initially, the reward function is

$$f'_0(\boldsymbol{y}) = f(\boldsymbol{y}) = \sum_{u_i \in \mathcal{U}} \sum_{j=1}^{m} \sum_{k=1}^{l_j} c_{ijk}^{(0)'} y_{ijk},$$

where $c_{ijk}^{(0)'} = c_{ijk}$. In each iteration I, we first select m sensing tasks from total m' sensing tasks in the following way: for each j, find the task t_{jk_j}, such that

$$k_j = \arg \max_{k=1,2,\cdots,l_j} c_{Ijk}^{(I-1)'}.$$

Denote the selected sensing tasks by $\mathcal{T}^{(I)}$. Then solve the orienteering problem associated with user u_I, which can be similarly modeled as in Sec. III-A by a graph $G_I(V_I, E_I)$. Hereby, $V_I = u_I \cup \mathcal{T}^{(I)}$, E_I denotes all the edges connecting any two vertices in V_I and the weight of any edge in E_I is the distance of its two end vertices. Vertex u_I is the root vertex and its reward is 0. The reward of vertex t_{jk_j} is $c_{Ijk_j}^{(I-1)'}$. Denote by O_I the assignment of sensing tasks to user u_I that is obtained by an algorithm of orienteering problem.

We now proceed to define a new reward function $f_I(\boldsymbol{y})$ at iteration I,

$$f_I(\boldsymbol{y}) = \sum_{u_i \in \mathcal{U}} \sum_{j=1}^{m} \sum_{k=1}^{l_j} c_{ijk}^{(I)} y_{ijk},$$

where $c_{ijk}^{(I)}$ is given by

$$c_{ijk}^{(I)} = \begin{cases} c_{Ijk}^{(I-1)'}, \text{ if } i = I, \text{ or } i > I, \text{ and } t_{jk} \in O_I \\ 0, \text{ otherwise.} \end{cases} \tag{1}$$

Then let $f'_I(\boldsymbol{y}) = f'_{I-1}(\boldsymbol{y}) - f_I(\boldsymbol{y})$, i.e., the reward of (i, j, k) in $f'_I(\boldsymbol{y})$ is given by $c_{ijk}^{(I)'} = c_{ijk}^{(I-1)'} - c_{ijk}^{(I)}$. The process terminates when $I = m$ or $f'_I(\boldsymbol{y}) = 0$. Note that when $I = m$, it is obvious that $c_{ijk}^{(I)} = c_{ijk}^{(I-1)'}$, which leads to $f'_{n-1}(\boldsymbol{y}) = f_n(\boldsymbol{y})$ and $f'_n(\boldsymbol{y}) = 0$.

3) Refining the final assignment backwards. Note that in the second process, each sensing task t_{jk} may be allocated to multiple users at different iterations. In this step, LRBA tries to refine the final assignment based on the results of step 2). As $c_{ijk}^{(I)'} = c_{ijk}^{(I-1)'} - c_{ijk}^{(I)}$, the value of each reward $c_{ijk}^{(I)'}$ decreases with the increase of iteration I. Task t_{jk} with $c_{ijk}^{(I)'} < 0$ will never be selected in the *orienteering problem*. In step 2), if task t_{jk} selected by the algorithm of *orienteering* problem at iteration I_1 is selected again at iteration $I_2, I_2 > I_1$, this means the original c_{I_2jk} is high since $c_{I_2jk}^{(I_2-1)'}$ is relatively high such that t_{jk} is selected by the algorithm of *orienteering problem* at iteration I_2. Hence, assigning task t_{ij} to I_2 may be more beneficial than the case when assigning task t_{jk} to I_1. In view of this, the step 3) refines the final sensing task assignment backwards, i.e., allocate sensing tasks to users from u_n to u_1. Denote the final task assignment of user I by S_I. At the iteration I, for each sensing task $t_{jk} \in O_I$, if it has been allocated to user $u_{I'}, I' > I$, then it will be excluded from S_I, i.e., $S_I = O_I \setminus \cup_{k=I+1}^{n} S_k$. The final task assignment then is $\mathcal{S} = \{S_1, S_2, \cdots, S_n\}$.

D. An illustrative example

In this subsection, we give a simple example to illustrate how LRBA works.

We set $n = 3, m = 4$, and $l_j = 3, \forall\, j = 1, 2, 3, 4$. The locations of 3 users and 4 sensing tasks are randomly selected on an area of $15m \times 15m$, as illustrated in Fig. 1(a). The budget B_i of each user i is set randomly from $[10, 20]$, and the reward R_{ij} is set randomly from $[3, 6]$. As $l_j = 3$, LRBA first transforms the original MRP into N-MRP. We illustrate the reward c_{ijk} of function $f(\boldsymbol{y})$ of N-MRP in Fig. 1(b), where each sensing task t_j is divided into 3 independent tasks $t_{jk}, k = 1, 2, 3$, each having the same reward as t_j.

At the second step, the LRBA allocates sensing tasks to each user iteratively from 1 to 4. When allocating tasks to user 1, LRBA first chooses the tasks t_{11}, t_{21}, t_{31} and t_{41} as t_{jk}, $k = 1, 2, 3$, have the same reward for each j, and we choose the task with smaller k. Then LRBA adopts algorithm of *orienteering* problem to allocate tasks $\{t_{31}, t_{21}, t_{11}\}$ to user 1, as illustrated in Fig. 1(a) by the red arrow starting from user

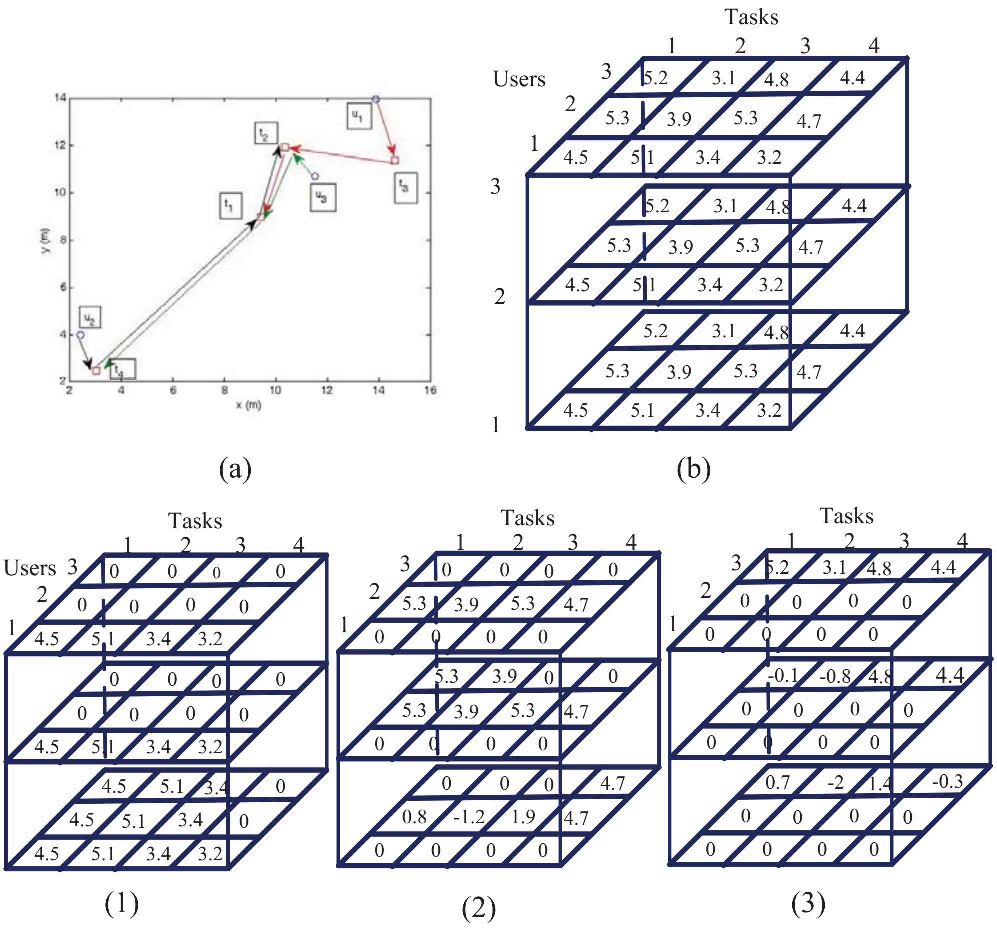

Fig. 1. An illustration of LRBA.

1. After the allocation of tasks to user 1, LRBA can thus decide the reward function $f_1(\boldsymbol{y})$, i.e.,

$$\begin{cases} c_{1jk}^{(1)} = c_{1jk}, j = 1,2,3,4, k = 1,2,3 \\ c_{ij1}^{(1)} = c_{1j1}, j = 1,2,3, i = 2,3 \\ c_{ijk}^{(1)} = 0, \quad \text{otherwise}, \end{cases} \tag{2}$$

which is shown in Fig. 1-(1). In the above Eq. (2), $c_{1jk}^{(1)} = c_{1jk}$ for user u_1, and $c_{ij1}^{(1)} = c_{1j1}$ for $j = 1,2,3, i = 2,3$, as $\{t_{11}, t_{21}, t_{31}\}$ is selected by the algorithm of *orienteering* problem for user u_1. With reward function $f_1(\boldsymbol{y})$, we can have function $f_1'(\boldsymbol{y}) = f(\boldsymbol{y}) - f_1(\boldsymbol{y})$. Similarly, LRBA allocates sensing tasks $\{t_{41}, t_{12}, t_{22}\}$ to user 2 and $\{t_{23}, t_{13}, t_{42}\}$ to user 3. The reward functions $f_2(\boldsymbol{y})$ and $f_3(\boldsymbol{y})$ are illustrated in Fig. 1-(2) and Fig. 1-(3), respectively. Note that $f(\boldsymbol{y}) = f_1(\boldsymbol{y}) + f_2(\boldsymbol{y}) + f_3(\boldsymbol{y})$.

At the third step, LRBA refines the allocation backwards. In this example, no task t_{jk} is allocated to more than one users. Hence, LRBA does nothing. The allocation in the second step is the final allocation for all users. If t_{jk} is allocated to user 2 and 3 in the second step, LRBA only allocates t_{jk} to user 3 in the third step.

IV. Theoretical Analysis

In this section, we investigate the theoretical performance of LRBA, and show that LRBA is a 5-approximate algorithm to MRP. We begin with introducing some useful Lemmas.

Lemma 1: $\mathcal{S}$ is a feasible solution of N-MRP.

Proof: From step 2), a user u_i can be assigned with at most one sensing task from $\{t_{jk}, k = 1,2,\cdots,l_j\}$ for a fixed j. By adopting an existing *orienteering* algorithm, O_i satisfies the traveling distance budget, and so does $S_i \subseteq O_i$ (since Euclidian distance metric satisfies triangle inequality). From step 3), each sensing task can be allocated to at most one user. Therefore, $\mathcal{S}$ is a feasible solution of N-MRP. ■

In the step 1), we transform MRP to N-MRP. The following result shows MRP and N-MRP are essentially equivalent.

Lemma 2: For any feasible solution $\boldsymbol{x}$ of MRP $(\mathcal{C}, g(\boldsymbol{x}))$, we can find a corresponding feasible solution $\boldsymbol{y}$ of N-MRP $(\mathcal{F}, f(\boldsymbol{y}))$ such that $f(\boldsymbol{y}) = g(\boldsymbol{x})$; and vice verse.

Proof: Let $\mathcal{U}_j = \{u_i | x_{ij} = 1\}$, i.e., the set of users that are assigned to perform sensing task t_j. Sort $\mathcal{U}_j$ in an increasing order according to their ID. Let k_i be the order of user u_i in the sorted set $\mathcal{U}_j$. For any feasible $\boldsymbol{x}$, set $\boldsymbol{y}$ as follows: for each sensing task t_j, if $u_i \in \mathcal{U}_j$, let $y_{ijk_i} = x_{ij}$, otherwise $y_{ijk} = 0$, $k = 1, 2, \cdots, l_j$. It is obvious that $\boldsymbol{y}$ is a feasible solution of N-MRP. In addition, we have $\sum_{k=1}^{l_j} y_{ijk} = x_{ij}$, which is because $\sum_{k=1}^{l_j} y_{ijk} = 1 = x_{ij}$ if $u_i \in \mathcal{U}_j$ and $\sum_{k=1}^{l_j} y_{ijk} = 0 = x_{ij}$ if $u_i \notin \mathcal{U}_j$. Note that $c_{ijk} = R_{ij}, \forall k = 1, 2, \cdots, l_j$. We obtain that

$$f(\boldsymbol{y}) = \sum_{u_i \in \mathcal{U}} \sum_{j=1}^{m} \sum_{k=1}^{l_j} c_{ijk} y_{ijk} = \sum_{u_i \in \mathcal{U}} \sum_{j=1}^{m} R_{ij} x_{ij} = g(\mathbf{x}).$$

For any feasible solution $\boldsymbol{y}$ of N-MRP, let $x_{ij} = \sum_{k=1}^{l_j} y_{ijk}$. In a similar way, we can show that $\boldsymbol{x}$ is a feasible solution of MRP and $f(\boldsymbol{y}) = g(\boldsymbol{x})$, which completes the proof. ■

Consider a specific case of N-MRP problem when there is only one mobile user u_I, which is expressed by

$$N - MRP_I : \max \sum_{j=1}^{m} \sum_{k=1}^{l_j} c_{Ijk}^{(I-1)'} y_{Ijk}$$

$$\begin{cases} \sum_{k=1}^{l_j} y_{Ijk} \leq 1, \; j = 1, 2, \cdots, m. \\ D(\mathcal{P}(\mathcal{T}_I')) \leq B_I \,. \\ y_{Ijk} = \{0, 1\}. \end{cases}$$

The orienteering problem associated with each user u_I in the step 2) (denoted by OP$_I$) is a little different from N-MRP$_I$ in that all the sensing tasks are considered in N-MRP$_I$ while only m sensing tasks are selected for consideration in OP$_I$. Let $\boldsymbol{y}^*$ be the optimal solution of N-MRP$_I$. For all tasks t_{jk} satisfying $y_{Ijk}^* = 1$, there exists a task $t_{jk_j}, t_{jk_j} \in \mathcal{T}^{(I)}$ such that $c_{Ijk_j}^{(I-1)'} \geq c_{Ijk}^{(I-1)'}$. This is true since from the selection of $\mathcal{T}^{(I)}$, $c_{Ijk_j}^{(I-1)'}$ has the largest value among all $c_{Ijk}^{(I-1)'}, k = 1, 2, \cdots, l_j$. Since $\sum_{k=1}^{l_j} y_{Ijk}^* \leq 1$, we have

$$\sum_{j=1}^{m} \sum_{k=1}^{l_j} c_{Ijk}^{(I-1)'} y_{Ijk}^* \leq \sum_{j=1}^{m} \max_k c_{Ijk}^{(I-1)'} = \sum_{j=1}^{m} c_{Ijk_j}^{(I-1)'},$$

from which it is easy to conclude that N-MRP$_I$ has the same maximum total rewards as that of OP$_I$. Denote by β the approximate ratio of an existing algorithm for the orienteering problem. Hence, as O_I is a β-approximate solution of OP$_I$, it is also a β-approximate solution of N-MRP$_I$.

Lemma 3: For the reward functions in the step 2), we have

$$f(\boldsymbol{y}) = f_1(\boldsymbol{y}) + f_2(\boldsymbol{y}) + \cdots + f_{n-1}(\boldsymbol{y}) + f_n(\boldsymbol{y}).$$

Proof: From step 2), we have

$$\begin{aligned} f_1'(\boldsymbol{y}) &= f_0'(\boldsymbol{y}) - f_1(\boldsymbol{y}) \\ f_2'(\boldsymbol{y}) &= f_1'(\boldsymbol{y}) - f_2(\boldsymbol{y}) \\ \cdots &= \cdots \\ f_{n-1}'(\boldsymbol{y}) &= f_{n-2}'(\boldsymbol{y}) - f_{n-1}(\boldsymbol{y}) \\ 0 &= f_{n-1}'(\boldsymbol{y}) - f_n(\boldsymbol{y}) \end{aligned}$$

It is easy to get that

$$f_{I-1}'(\boldsymbol{y}) = f_I(\boldsymbol{y}) + f_{I+1}(\boldsymbol{y}) + \cdots + f_n(\boldsymbol{y}),$$

$I = 1, 2, \cdots, n$, which, combined with the fact that $f_0'(\boldsymbol{y}) = f(\boldsymbol{y})$, completes the proof. ■

Recall that $\mathcal{S} = \{S_1, S_2, \cdots, S_n\}$ is the final assignment. With Lemma 3, we will show in the following Lemma that $\mathcal{S}$ is actually a $(1+\beta)$-approximate solution to problem $(\mathcal{F}, f_I(\cdot))$.

Lemma 4: $\mathcal{S}$ is a $(1+\beta)$-approximate solution to problem $(\mathcal{F}, f_I(\cdot))$.

Proof: When $I = n$, we have $f_{n-1}'(\boldsymbol{y}) = f_n(\boldsymbol{y})$. For the reward function $f_n(\boldsymbol{y})$, $c_{ijk}^{(n)} = 0, \forall i = 1, 2, \cdots, n-1$. To maximize $(\mathcal{F}, f_n(\boldsymbol{y}))$ is equivalent to maximize N-MRP$_I$ with $I = n$. Therefore, $S_n = O_n$ is a β-approximate solution of N-MRP$_n$. As $S_n \subset \mathcal{S}$, this leads to the fact that $\mathcal{S}$ is a β-approximate (thus also $(1+\beta)$-approximate) solution of $(\mathcal{F}, f_n(\boldsymbol{y}))$.

We now show when $I < n$, $\mathcal{S}$ is $(1+\beta)$-approximate solution of $(\mathcal{F}, f_I(\boldsymbol{y}))$. Note that for reward function $f_I(\boldsymbol{y})$, $c_{ijk}^{(I)} = 0$ if i) $i < I$ or ii) $i > I$ and $t_{jk} \notin O_I$. We can rewrite $f_I(\boldsymbol{y})$ as

$$\begin{aligned} f_I(\boldsymbol{y}) &= \sum_{i \in \mathcal{U}} \sum_{j=1}^{m} \sum_{k=1}^{l_j} c_{ijk}^{(I)} y_{ijk} \\ &= \sum_{j=1}^{m} \sum_{k=1}^{l_j} c_{Ijk}^{(I)} y_{Ijk} + \sum_{i>I} \sum_{t_{jk} \in O_I} c_{ijk}^{(I)} y_{ijk} \\ &= \sum_{j=1}^{m} \sum_{k=1}^{l_j} c_{Ijk}^{(I-1)'} y_{Ijk} + \sum_{i>I} \sum_{t_{jk} \in O_I} c_{Ijk}^{(I-1)'} y_{ijk} \end{aligned} \tag{3}$$

The third equation holds from Eq. (1). Obviously, the first term of Eq. (3) is the objective function of N-MRP$_I$. Denote the total reward of allocation O_I by $R(O_I)$. Since the algorithm of orienteering problem has a β approximate ratio, we have

$$\sum_{j \in \mathcal{T}} \sum_{k=1}^{l_j} c_{Ijk}^{(I)} y_{Ijk} \leq \beta R(O_I).$$

For the second term of Eq. (3), the reward of $(i, j, k), i > I, t_{jk} \in O_I$, is the same as the reward of (I, j, k). The total reward of the second term under any feasible solution is thus at most $R(O_I)$. We have

$$f_I(\boldsymbol{y}) \leq \beta R(O_I) + R(O_I) = (1+\beta) R(O_I), \tag{4}$$

which means that O_i is a $(1+\beta)$-approximate solution of $(\mathcal{F}, f_I(\cdot))$.

In the final assignment $\mathcal{S}$, it is obvious that, for some task t_{jk}, it satisfies one of the following two conditions: 1) $t_{jk} \in O_I$ and $t_{jk} \in S_I$, or 2) $t_{jk} \in O_I$ and $t_{jk} \in S_{I'}$, where $I' > I$. For the case 2), we know that $c_{Ijk} = c_{I'jk}$ as t_{jk} is selected at iteration I. Then from Eq. (4) we get

$$f_I(\boldsymbol{y}) \leq (1+\beta) R(O_I) \leq (1+\beta) R(\mathcal{S}).$$

Therefore, $\mathcal{S}$ is a $(1+\beta)$-approximate solution of $(\mathcal{F}, f_I(\cdot))$.

■

With all the preparations, now we give the main result about the performance of LRBA.

Theorem 3: LRBA is a $(1+\beta)$-approximate algorithm for MRP.

Proof: According to Lemma 2, we only have to show that LRBA is a $(1+\beta)$-approximate algorithm for N-MRP, which is true from Lemma 3, Lemma 4 and local ratio theorem. ■

From [14], the best approximate ratio of an *orienteering* algorithm is 4. With Theorem 3, we get that LRRA is a 5-approximate algorithm.

V. Pricing Sensing Tasks

In the previous sections, $R_{ij} = r'_{ij} - r_{ij}$ is given and we focus on the allocation problem. In this section, we discuss how to determine the price of each sensing task for each mobile user.

Recall that r'_{ij} is the reward that the platform will receive for allocating task t_j to user u_i. We further denote by r''_{ij} the cost for mobile user u_i to perform sensing task t_j. We consider $r'_{ij} \geq r''_{ij}$, as otherwise there is no incentive for mobile user u_i to perform task t_j. Based on the cost r''_{ij}, user u_i negotiates with the platform about the price r_{ij} of sensing task t_j. In this section, we employ bargaining theory to design a fair pricing mechanism.

Bargaining theory is a desirable tool for tasks pricing in *crowdsensing*. In bargaining theory, two or more players bargain to reach an agreement from which all can benefit. A very important concept in bargaining theory is the Nash bargaining solution, which is known to be the only solution that satisfies the following four axioms (please refer to [17] for the details).

- Invariants to equivalent utility representations,
- Pareto efficiency,
- Symmetry,
- Independence of irrelevant alternatives.

We first consider the case when there is only one mobile user u_i in the platform. The rewards gained by the platform and mobile user u_i are both 0, if they fail to reach an agreement. Therefore, for an agreement on price r_{ij}, $r'_{ij} - r_{ij}$ and $r_{ij} - r''_{ij}$ are the potential gains for the platform and mobile user u_i, respectively. The Nash bargaining solution [17] provides a fair price for both the platform and the mobile user u_i, which can be obtained by solving the following problem.

$$\begin{aligned} P_1: \quad & \max\,(r_{ij} - r''_{ij})(r'_{ij} - r_{ij}) \\ & s.t.\, r''_{ij} \leq r_{ij} \leq r'_{ij} \end{aligned}$$

For the case with only one mobile user, it is easy to see that $r_{ij} = \frac{r'_{ij}+r''_{ij}}{2}$, indicating that when there is only one mobile user, the platform has to share the same reward with mobile user. Otherwise, the mobile user may feel unfair and reject the agreement, in which case the platform could gain nothing.

Next, we consider the case of multiple users. The outcomes of multiple players case in bargaining theory depend on the sequence of bargaining and whether or not there is coalition, which can be extremely complex in some scenarios. In this work, we try to employ bargaining theory to obtain a desirable pricing, and the task allocation is not included in the pricing process [3]. Hence, we adopt a simple but insightful model: mobile users do not form coalition; the platform bargains with each mobile user and the agreement on one mobile user will not impact the agreement on other users (since this user may not be chosen by the platform in the allocation step).

Assume that there are n' users who are interested in sensing task t_j [4]. If the platform and user u_i fail to reach an agreement, the reward of user u_i is still 0. However, the platform can have chance to bargain with other $n'-1$ users and thus the expected reward can be larger than 0. Notice that when $r_{ij} \geq \frac{r'_{ij}+r''_{ij}}{2}$, mobile user u_i will definitely accept the agreement, as this is no less than what u_i can get even in the single user case. When $r_{ij} < \frac{r'_{ij}+r''_{ij}}{2}$, the probability that mobile user u_i would accept the agreement can be basically characterized by

$$p = \frac{r_{ij} - r''_{ij}}{\frac{r'_{ij}+r''_{ij}}{2} - r''_{ij}} = \frac{2(r_{ij} - r''_{ij})}{r'_{ij} - r''_{ij}}.$$

p indicates that the willingness of user u_i to accept the agreement depends on the ratio of current reward to the maximum possible reward. The platform can use p to calculate the probability that other than user u_i there is at least one mobile user accepting the agreement, which is $1-(1-p)^{n'-1}$. Hence, the expected reward is $(r'_{ij} - r_{ij})(1-(1-p)^{n'-1})$. The gain of the platform to reach the agreement on r_{ij} is then

$$(r'_{ij} - r_{ij}) - (r'_{ij} - r_{ij})(1-(1-p)^{n'-1}) = (r'_{ij} - r_{ij})(1-p)^{n'-1}.$$

Similarly, we can solve the following maximum problem to find the Nash bargaining solution.

$$\begin{aligned} P_2: \quad & \max\,(r_{ij} - r''_{ij})(r'_{ij} - r_{ij})(1-p)^{n'-1} \\ & s.t. \quad r''_{ij} \leq r_{ij} \leq r'_{ij} \end{aligned}$$

It is easy to show that the problem P_2 is equivalent to the following problem:

$$\begin{aligned} P_3: \quad \max \psi(r_{ij}) \;&=\; \log(r_{ij} - r''_{ij}) + \log(r'_{ij} - r_{ij}) \\ &\quad + (n'-1)\log\left(\frac{r'_{ij} + r''_{ij} - 2r_{ij}}{r'_{ij} - r''_{ij}}\right). \end{aligned}$$

We can get that $\frac{d^2\psi(r_{ij})}{dr_{ij}^2} < 0$. By solving

$$\frac{d\psi(r_{ij})}{dr_{ij}} = \frac{1}{r_{ij} - r''_{ij}} - \frac{1}{r'_{ij} - r_{ij}} - \frac{2(n'-1)}{r'_{ij} + r''_{ij} - 2r_{ij}} = 0,$$

we can obtain the optimal solution of $P3$,

$$r_{ij} = \frac{r'_{ij} + r''_{ij} - \sqrt{\frac{n'-1}{n'+1}}(r'_{ij} - r''_{ij})}{2}.$$

Remarks. When $n' = 1$, $r_{ij} = \frac{r'_{ij}+r''_{ij}}{2}$ reduces to the single user case; when $n' \to \infty$, $r_{ij} \to r''_{ij}$. This indicates that, when

[3] See the allocation problem in Sec. III.

[4] $n' \leq n$ as some mobile users may be not interested in task t_j.

market demand increases (i.e., more users want to claim the sensing task), the platform can bring down the cost of sensing task performing, which is similar to the real market.

We illustrate how to attain a simple but efficient pricing mechanism. In specific crowdsensing applications, this simple model can be modified to characterize more complex situations.

VI. Numerical Results

In this section, we provide simulation results to show some properties and advantages of LRBA. All the simulations are performed using MATLAB.

The simulation settings are as follows. The parameters in the simulations are set in a random manner. The locations of m sensing tasks and n users are randomly selected in an area of $30\,unit \times 30\,unit$. We will vary the values of m and n to show the performance of LRBA under different scenarios. The distance between any two locations is calculated by the Euclidian distance between them. The time budget of each user (in term of the total distance) is a random variable $\zeta + \nu$, where ζ is a constant and ν is uniformly distributed in the interval $[0, 5]$. $l_j, j = 1, 2, \cdots, m$, is uniformly distributed in $[1, 4]$. The platform can gain a reward R_{ij} if sensing task t_j is allocated to user u_i, which is also a number uniformly distributed in $[1, 6]$. We also use other settings of parameters and the results are consistent with what we present in this section.

We first vary the number of sensing tasks m from 35 to 65 while the number of users n is fixed to 15. ζ is set to be 10. The total rewards obtained by LRBA for each scenario are recorded and the results are plotted in Fig. 2. Clearly, when the number of sensing tasks increases, the total rewards also increase. The total number of tasks performed by each user may not increase, since the traveling time budget of each user does not change. However, each user has more choices and thus has more chances to perform sensing tasks with higher reward, as the total number of sensing tasks becomes larger. Therefore, the total rewards increase as the number of sensing tasks increases. We also set n to be 20 and 25, and perform the simulations respectively. The results are also plotted in Fig. 2, which are consistent with the results when $n = 15$.

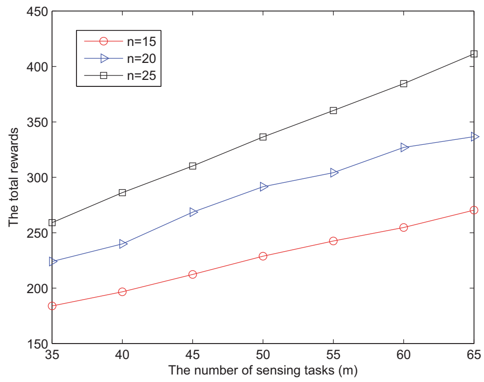

Fig. 2. The total rewards obtained by LRBA Vs different number of sensing tasks m when n is fixed to 15, 20, 25, respectively.

We then perform simulations to show the impact of total number of user n. Note that when $m > n$, the total reward will obviously increase as the number of users n increases (since more tasks will be performed as n increases). To this end, we let $m \leq n$, and vary the number of users n from 35 to 65 while the number of tasks m is fixed to 30, 35, 40, respectively. ζ is again set to be 10. The results for each scenario are plotted in Fig. 3. As we can see, the total reward is still an increasing function of n, which is because more users offers more opportunities to allocate the sensing tasks to mobile users with higher rewards. In this way, though the number of tasks does not increase, higher total rewards can be achieved by allocating tasks more efficiently.

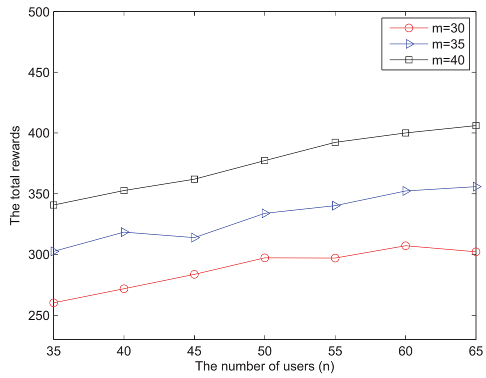

Fig. 3. The total rewards obtained by LRBA Vs different number of users n when m is fixed to 30, 35, 40, respectively.

Finally, we show the advantage of LRBA. As there is no existing algorithm for our problem, we compare LRBA with the well-known greed algorithm (GA), which can obtain desirable results in many set cover and packing based problems. In GA, the original MRP is also transformed to N-MRP as the step 1 of LRBA. Then GA solves the N-MRP greedily, i.e., for user i, GA uses existing orienteering algorithm to allocate sensing tasks, and once the task is allocated to a user, it can not be allocated to other users again. In GA, there is no dynamic reward function at each iteration and backward refinement step, which are included in LRBA. We vary the number of sensing tasks m from 35 to 65 while the number of users n is fixed to 15, 20 and 25, respectively. We plot the results in Fig. 4. It is clear that LRBA outperforms GA, as for each scenario, LRBA can obtain the total reward two times of that obtained by GA. This shows the significant advantage of LRBA over GA.

VII. Related Work

Recent years, *crowdsensing* has received great attention due to the advance in MEMS and wireless communications [18]–[22]. Most existing work focused on designing a system for a specific sensing application. Hull *et al.* [7] proposed a platform named CarTel, which consists of a collection of CarTel nodes, a CalNet and a Portal. A CarTel node is a small-sized computer embedded with a set of sensors such as camera and GPS. The CalNet is a delay tolerant network formed by CarTel

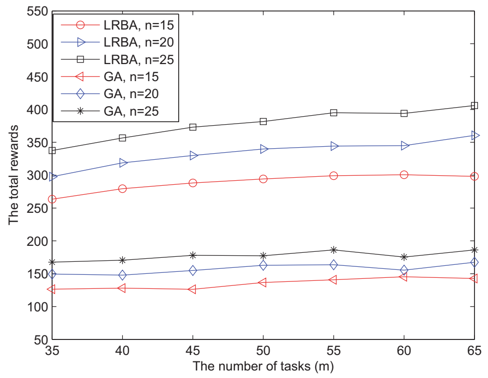

Fig. 4. The total rewards obtained by LRBA and greedy algorithm (GA) Vs different number of sensing tasks m when n is fixed to 10, 15, 20, respectively.

nodes, and the Portal is a backend data center for data fusion and analysis. The CarTel nodes are mounted on a mobile vehicle and collect traffic related information such as images and vehicle speed. The CarTel node can upload the sensory data to the Portal via open access networks (e.g., Wifi) or CalNet when available. As discussed in Sec. I, Dutta *et al.* [6] proposed *Common Sense* to monitor the air quality level dynamically in a city. Reddy *et al.* [8] proposed a framework, ImageScape, leverage smartphones to collect image and audio data of daily lives. Chon *et al.* [3] presented a framework, CrowdSense@Place (CSP), to understand the semantics of places, using the data that a typically commercial smartphone can collect. Chen *et al.* [5] designed a crowdsensing platform to find available parking on the street. Please refer to [9], [10], [12] for more recent works.

VIII. Conclusion

In this paper, we studied the problem of allocating location dependent tasks in *crowdsensing* applications. We first focused on the task allocation problem and showed that the problem is NP hard and proposed an efficient algorithm, LRBA, which divides the whole problem into several subproblems by modifying the reward function at each iteration. We investigated the theoretical result of LRBA, and obtained the approximate ratio of aggregate reward obtained by LRBA to that by the optimal. We then designed an efficient pricing mechanism for the platform and each mobile user to reach an agreement on the price of each task. Extensive simulation results were provided to demonstrate the efficiency of the proposed algorithms.

References

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

[2] A. Korthaus and W. Dai. Crowdsourcing in heterogeneous networked environmentscopportunities and challenges. In *Proceedings of the International Conference on Network-Based Information Systems*, 2012.

[3] Y. Chon, N. D. Lane, F. Li, H. Cha, and F. Zhao. Automatically characterizing places with opportunistic crowdsensing using smartphones. In *Proceedings of the International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2012.

[4] D. Yang, G. Xue, X. Fang, and J. Tang. Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing. In *Proceedings of The Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2012.

[5] X. Chen, E. Santos-Neto, and M. Ripeanu. Crowdsourcing for on-street smart parking. In *Proceedings of the ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet)*, 2012.

[6] P. Dutta, P. M. Aoki, N. Kumar, and A. Mainwaring. Demo abstract: Common sense: Participatory urban sensing using a network of handheld air quality monitors. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2009.

[7] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden. Cartel: A distributed mobile sensor computing system. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2006.

[8] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, and M. Hansen. Image browsing, processing, and clustering for participatory sensing: Lessons from a dietsense prototype. In *Proceedings of the Workshop on Embedded Networked Sensors (EmNets)*, 2007.

[9] M. Demirbas, M. A. Bayir, C. G. Akcora, and Y. S. Yilmaz. Crowd-sourced sensing and collaboration using twitter. In *Proceedings of IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, 2010.

[10] F. C. Pereira, A. Vaccari, F. Giardin, C. Chiu, and C. Ratti. *Crowdsensing in the Web: Analyzing the Citizen Experience in the Urban Space*, chapter From Social Butterfly to Engaged Citizen: Urban Informatics, Social Media, Ubiquitous Computing, and Mobile Technology to Support Citizen Engagement. Cambridge (MA): MIT Press, 2012.

[11] L. Yang, H. Kim, J. Zhang, M. Chiang, and C. W. Tan. Pricing-based decentralized spectrum access control in cognitive radio networks. *IEEE/ACM Transactions on Networking*, 21(2):522–535, 2013.

[12] H. S. Packer, S. Samangooei, J. S. Hare, N. Gibbins, and P. H. Lewis. Event detection using twitter and structured semantic query expansion. In *Proceedings of International Workshop on Multimodal Crowd Sensing (CrowdSens)*, 2012.

[13] M. Ra, B. Liu, T. L. Portay, and R. Govindan. Medusa: A programming framework for crowd-sensing applications. In *Proceedings of the International Conference on Mobile Systems, Applications and Services (Mobisys)*, 2012.

[14] A. Blum, S. Chawla, D. R. Karger, T. Lane, A. Meyerson, and M. Minkoff. Approximation algorithms for orienteering and discounted-reward tsp. *Journal SIAM Journal on Computing*, 37(2):653–670, 2007.

[15] R. Bar-Yehuda, K. Bendel, A. Freund, and D. Rawitz. Local ratio: a unified framework for approximation algorithms. *Journal ACM Computing Surveys (CSUR)*, 36(4):422–463, 2004.

[16] R. Cohen, L. Katzir, and D. Raz. An efficient approximation for the generalized assignment problem. *Information Processing Letters*, 100(4):162–166, 2006.

[17] A. Muthoo. *Bargaining Theory with Applications*. Cambridge University Press, Cambridge, U.K., 1999.

[18] L. He, Z. Yang, J. Pan, L. Cai, and J. Xu. Evaluating service disciplines for mobile elements in wireless ad hoc sensor networks. In *Proceedings of IEEE Conference on Compute Communications (INFOCOM)*, 2012.

[19] X. Cao, J. Chen, Y. Zhang, and Y. Sun. Development of an integrated wireless sensor network micro-environment monitoring system. *ISA Transactions(Elsevier)*, 47(3):247–255, 2008.

[20] A. Abdulla, H. Nishiyama, J. Yang, N. Ansari, and N. Kato. Hymn: A novel hybrid multi-hop routing algorithm to improve the longevity of wsns. *IEEE Transactions on Wireless Communications*, 11(7):2531–2541, 2012.

[21] D.-H. Shin and S. Bagchi. Optimal monitoring in multi-channel multi-radio wireless mesh networks. In *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*.

[22] S. He, J. Chen, D. Yau, H. Shao, and Y. Sun. Energy-efficient capture of stochastic events under periodic network coverage and coordinated sleep. *IEEE Transactions on Parallel and Distributed Systems*, 23(6):1090–1102, 2012.