Location-dependent Task Assignment for Opportunistic Mobile Crowdsensing

Fatih Yucel and Eyuphan Bulut
Department of Computer Science, Virginia Commonwealth University
401 West Main St. Richmond, VA 23284, USA
{yucelf, ebulut}@vcu.edu

***Abstract*—In mobile crowdsensing applications that rely on opportunistic sensing and communication, efficient task assignment strategies are needed to ensure that the tasks are completed before their expiration time. This requires to optimize the trade-off between high task completion ratio and cost-efficiency by assigning tasks only to a small group of users who are expected to be of most assistance to task owners. To address this issue, in this paper, we propose two new task assignment protocols based on a new metric that accurately measures the utility of users to each other in performing tasks in specific regions. Through simulations we show that the proposed protocols not only provide a high task completion ratio, but also utilize the network resources efficiently by assigning tasks to as few users as possible, hence they perform better than the previous work.**

***Index Terms*—Task assignment, crowdsensing, mobile social networks.**

I. Introduction

Widespread use of mobile devices and their advanced sensing capabilities have recently led to the emergence of a new paradigm called Mobile Crowdsensing (MCS) where users exploit the power of crowd to have their sensing tasks completed faster and without requiring personal effort (e.g., visiting sensing region to perform the task individually). The main challenging issue in MCS is the assignment of tasks to users in a way that maximizes the system utility (e.g., number of completed tasks, average task completion time) while using network resources efficiently.

Task assignment protocols in MCS are generally classified with respect to the involvement and activeness of users in performing tasks [1]. In *participatory sensing*, a user that is assigned to a task interrupts his mobility and goes to the corresponding sensing region to carry out the task, and then also needs to deliver the sensed data to the task owner. Therefore, the travel costs of users should be taken into consideration during the task assignment process. On the other hand, in *opportunistic sensing*, users do not interrupt their daily schedule. Instead, they perform assigned tasks only when they happen to be in the sensing region and deliver the sensed data to task owner in their first encounter after the completion of task. Thus, opportunistic sensing has the advantage of not introducing extra travel costs to perform tasks, but it may cause larger task completion times if not planned properly.

In this paper, we study the location-dependent task assignment problem in opportunistic crowdsensing in mobile social networks (MSN). Consider the scenario illustrated in Fig. 1.

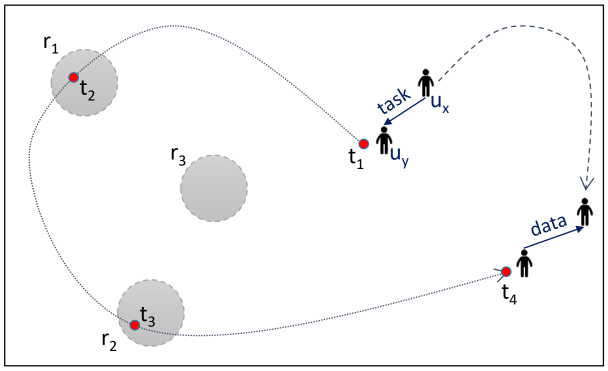

Fig. 1: An opportunistic crowdsensing instance with a task owner (u_x) who has sensing tasks to be performed at 3 different regions (i.e., r_1, r_2, and r_3). The mobility of u_x and u_y are, respectively, shown with dashed and dotted lines.

Assume that a user u_x has three sensing tasks that need to be performed at regions, r_1, r_2, and r_3, respectively. Instead of deviating from his usual route in order to go to these regions to individually perform the tasks, he starts a crowdsensing by requesting other users to carry out these tasks for him. However, due to the low range communication technologies used (e.g. Bluetooth, WiFi) and resulting intermittent connectivity in MSNs, it may not be possible to find users in these regions directly and communicate with them. Thus, an opportunistic approach is adopted. That is, when u_x encounters another user u_y in the network, he assesses the capability of u_y for completing and returning these tasks to him and decides on assigning them to u_y or not.

For example, in Fig. 1, when u_x and u_y encounter at t_1, assume u_x finds out that u_y can complete his tasks in r_1 and r_2 in a significantly shorter time than himself, thus forwards the related task information to u_y. Note that since he needs to pay a certain fee for each task he assigns to u_y, he tries to avoid assigning the tasks for which u_y would not be very helpful (e.g., the tasks in r_3). In fact, u_y visits the regions r_1 and r_2 before u_x does, and performs the sensing tasks at t_2 and t_3, respectively. Finally, when they meet at t_4, u_y delivers the sensed data to u_x.

Note that the problem described above differs from the routing problem in MSNs as it necessitates that the sensed data will be returned to the task owner, and differs from the classic task assignment problem in general MCS systems as the task assignments can only happen when nodes encounter opportunistically. In fact, we need to explore the following together to design an efficient task assignment process in the proposed system:

- The relationship between encounter patterns of nodes and their regional visit patterns, as each task is associated with a certain region.
- The ability of a node to not only carry out a task, but also to return the outcomes to the requester.

To this end, we define a new metric to measure the utility of a node u_y to a task owner u_x in completing the tasks in a region r_i and returning them back to u_x, and propose two assignment protocols based on this metric. Through simulations, we evaluate the performance of the proposed protocols in terms of various performance metrics such as the task completion ratio and the average completion time of tasks, and show that they perform better than the previous work.

The rest of the paper is organized as follows. In Section II, we present an overview of the related work. In Section III, we provide the system model and formally define the problem addressed in the paper. In Section IV, we present a new metric that measures utility of nodes for each other in performing tasks in different regions, and then propose two assignment protocols based on this metric. In Section V, we present evaluation of the proposed approaches through extensive simulations. Finally, we end up with conclusion in Section VI.

II. Related Work

In recent years, mobile crowdsensing has received a great attention and several aspects have been studied by many researchers. Besides the studies that focus on designing a system for a specific sensing application (e.g., image sharing [2], crowd GPS [3]), many studies [4]–[7] have looked at the user recruitment or task allocation problem in MCS systems. In these works, different objectives have been considered such as maximizing the number of completed tasks [7], minimizing the incentives provided to the users, assuring the task quality [6] and providing a secure user recruitment [8]. However, in these works, even though the location-dependent tasks are considered, the assignment process is mostly achieved in a deterministic and most of the time centralized manner (through cellular communication with users) considering the availability of users, budget constraints and sensing capabilities of user devices.

In a mobile crowdsensing system within a mobile social network, the short-distance wireless communication technologies (e.g., device-to-device (D2D), Bluetooth, Wi-Fi) between user devices are leveraged to achieve both the task allocation and data collection in an opportunistic fashion. Note that this reduces the overhead on cellular networks, and allows for local user recruitment and sensed data collection even if the cellular network coverage is poor [9], [10]. On the other hand, it comes with more challenging problems such as the task assignment problem in a localized, distributed and opportunistic manner. Recently, several studies have looked at these problems (e.g., task assignment and worker recruitment [11]–[13], incentive mechanism design [14], [15], and response and energy usage analysis [16]) and proposed solutions specific to opportunistic crowdsensing in mobile social networks. While these studies leverage the opportunistic encounters of nodes for task assignment and communication between nodes, they do not consider location-dependent tasks and simply assume any sensing task that requires some processing. Different from these aforementioned works, in this paper, we are considering location-dependent sensing task assignment in a mobile social network environment where both the encounter patterns between nodes and visit patterns of nodes to the task locations should be considered simultaneously for an efficient task assignment protocol.

III. System Model

We assume a system model with a set of mobile users $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$ and a set of well-defined regions $\mathcal{R} = \{r_1, r_2, \ldots, r_R\}$. We define a sensing task with a tuple as $s = (u, r)$, where u is the task owner and r is the region where the task needs to be performed. If a user has a sensing task to be performed in a set of various regions, he will create a separate sensing task for each region and treat them individually. Sensing tasks considered in the system are simple tasks that do not require long sensing/processing time such as taking a picture of a scene and measuring noise pollution. However, each task has an expiration time before which it has to be performed.

We also assume that users communicate with each other using only low range communication technologies such as Bluetooth, and there is no centralized node or server that manages the task assignments in the system. Thus, when a user u has a sensing task $s = (u, r)$, he should either wait until his next visit to r to perform it himself or get help from the crowd by assigning it to a group of users when he encounters them (i.e., when they are in his communication range). Users are incentivized by virtual credits to participate in this crowdsensing process, which they can then use to recruit other users to perform their sensing tasks. However, note that neither the task owner nor the other users assigned with the task will interrupt their daily schedule to exclusively go to r to perform the task, but instead, they will be notified by the mobile device when they happen to be in r (i.e., *opportunistic sensing*). For this reason, there is no travel cost associated with the tasks in our system.

Consequently, the objective of a task owner in our system is to minimize the number of assignments to save from his virtual credits, while assigning his tasks to sufficient number of users to make sure they will be completed before their expiration time. Hence, utility-based assignment protocols are needed to optimize this trade-off and to obtain efficient task assignments.

IV. Proposed Solution

The proposed solution has two parts. We first develop a new metric that can accurately quantify the utility of a user to

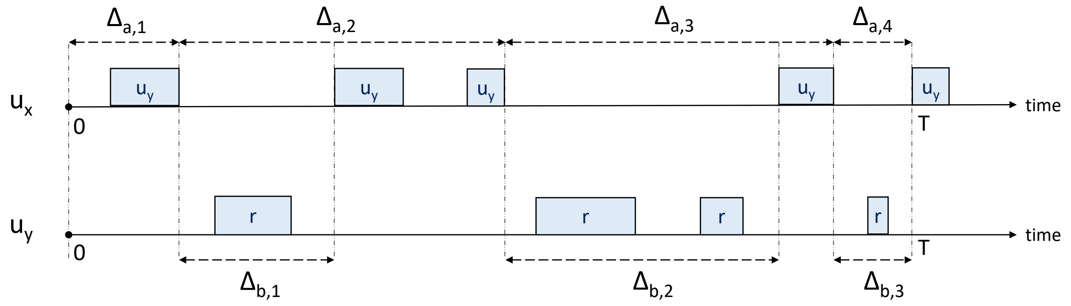

Fig. 2: An example encounter history of users u_x and u_y (upper timeline), and the corresponding visit pattern of node u_y to the region r (lower timeline).

another user for completing tasks and returning their outcomes back to the task owner. Then, we propose two task assignment algorithms utilizing this metric.

A. Expected Task Completion Time

In order to calculate the expected task completion time each user can offer, we need to understand not only the encounter relations between nodes but also their visiting patterns within task regions. In the literature of routing algorithms proposed for mobile social networks [17]–[19], it has been shown that social network metrics (e.g., friendship [19]) can be utilized to model the relations between nodes accurately. However, these cannot be applied directly to our problem as we need to consider the visiting patterns of nodes to task regions within their encounter patterns. Thus, we propose a new metric that can calculate the expected task completion time based on the historical encounter and visit patterns of nodes in the network.

To this end, we assume that each user had a simulated sensing task at every moment in the past for each possible region and compute the average completion time of each task in *two scenarios:* (i) when the task owner individually performs the task in the next visit to the region; (ii) when the task owner forwards the task to another user within the network, which will carry out the task in her next visit (v_{next}) to the region and then deliver the sensed data to the task owner in their first encounter after v_{next}.

In the *first scenario*, assume a task owner u_x would like to calculate the average time that it would take him to individually perform a task in region r. To this end, it generates a simulated sensing task s_t at each time unit t in a time frame of size T (i.e., $[0,T)$) in the past and finds its completion time based on his visit history to r in this time frame. Since there is not an additional delay due to delivery of the sensed data back to the task owner, the completion time $c(s_t)$ of a simulated sensing task s_t can simply be computed as:

$$c(s_t) = \begin{cases} t_{next} - t, & \text{if completed until } T \\ T - t, & \text{otherwise} \end{cases} \tag{1}$$

where t_{next} is the starting time of u_x's next visit to r, and $t_{next} = t$ if u_x is already in r, thus, $c(s_t) = 0$. Note that we have two assumptions in (1). The first one is that s_t has a minimal sensing/processing time requirement, so u_x is always able to complete the task no matter how short his visit to the region is. This covers many possible sensing tasks in practice such as taking pictures of an area which does not last more than a few seconds. The formulations however, could be updated easily to take into account longer processing times. Second, in order to reflect the negative impact of the tasks that are not completed by the end of time T (i.e., no visit to r happened after the task's generation at time t) on the average task completion time, we add the maximum possible delay in the considered time frame (i.e., $T - t$) to the calculations. Based on (1), u_x's average completion time of his own tasks in r can be calculated by:

$$\mathcal{C}^r_{u_x} = \frac{\int_0^T c(s_t)dt}{T} = \frac{1}{2T}\sum_{i=1}^{n} \Delta_i^2 \tag{2}$$

where n is the number of time intervals in which u_x was not in r until T (e.g., $n = 5$ for u_y in the lower timeline of Fig. 2), and Δ_i is the duration of ith interval.

In the *second scenario*, u_x would like to know the average time that it would take another user (u_y) he encountered to complete his task in region r and return the sensed data back to him. However, this is more complicated than the previous case given that both the encounter history of u_x and u_y, and the visit frequency of u_y in r should be taken into account. Consider the scenario given in Fig. 2. To be able to calculate the completion time of u_x's continuously generated sensing tasks s_t at each time unit t by u_y via a locally computable closed form function, we need to define two stages, namely stage a and b, as follows.

- *Stage a*: Starts at the end of the last encounter of u_x and u_y before u_y's next visit to r, and ends where the next stage a starts.
- *Stage b*: Starts when the corresponding stage a ends, and ends at the beginning of the next encounter of u_x and u_y. That is, the sensing tasks u_x generates during ith stage a (denoted by $\Delta_{a,i}$) are carried by u_y during ith stage b (denoted by $\Delta_{b,i}$), and completed and returned to u_x.

Therefore, in this second scenario, the completion time $c(s_t)$ of each sensing task s_t created at time t is:

$$c(s_t) = \begin{cases} \Delta_{a,i} + \Delta_{b,i} - l(t), & \text{if completed until } T \\ T - t, & \text{otherwise} \end{cases}$$

where $l(t) \in [0, \Delta_{a,i}]$ is the local time unit within ith stage a corresponding to the task generation time t. Then, the average completion time of user u_x's sensing tasks in r by user u_y is:

$$\begin{aligned} \mathcal{C}^r_{(u_x,u_y)} &= \frac{\int_0^T c(s_t)dt}{T} \\ &= \frac{1}{2T}\sum_{i=1}^{n}(2\Delta_{a,i}\Delta_{b,i} + \Delta_{a,i}^2) \end{aligned} \tag{3}$$

where n is the number of stage a instances until T, and $\Delta_{b,i} = 0$ if the tasks created in interval $\Delta_{a,i}$ are not yet completed.

B. Task Assignment Protocols

Note that users do not need to store their encounter or visit histories to calculate (2) or (3) as all they need are the updated values for the duration between last two visits in r for (2) and the duration of the last a and b stages for (3). Thus, a user u_x can maintain $\mathcal{C}^r_{u_x}$ values for each region r himself. However, the responsibility to update $\mathcal{C}^r_{(u_x,u_y)}$ and notify u_x during encounters will be on u_y's end as he is the one that has the records of both his encounter history with u_x and his visit history to r, so he can easily update $\mathcal{C}^r_{(u_x,u_y)}$ without requiring any data from u_x[1].

Once the values of the proposed metrics are known, a task owner u_x should decide whether to assign any subset of his tasks to the encounter user u_y. To this end, we propose two different assignment protocols, namely *Superiority-based* and *Best-K*.

- *Superiority-based*: u_x will assign his task $s = (u_x, r)$ to u_y only if u_y is likely to complete it *significantly* faster than himself. More formally, u_x assigns s to u_y if
$$\mathcal{C}^r_{u_x} > \alpha\mathcal{C}^r_{(u_x,u_y)}$$
where $\alpha > 1$ is a constant defined by u_x.
- *Best-K*: u_x first identifies a set of K users (e.g., u_z) with the lowest $\mathcal{C}^r_{(u_x,u_z)}$ values in the network, and assigns s to u_y only if the encountered node u_y is one of them. Hence, Best-K assignment protocol has a built-in parameter (K) that limits the number of assignments.

However, in general, we limit the number of assignments per task to L in both protocols, as the task owner will not benefit much from additional assignments after a certain point and the cost of additional task assignments will not be worthy.

V. Simulations

A. Settings

We generate a simulation environment using a community-based mobility model that has been used frequently in the literature ([19], [21]) particularly due to its ability to mimic periodicity in people's movements and encounters. In a 1000 m×1000 m area, we create $N = 100$ users and $R = 25$ non-overlapping regions of size 100 m×100 m which can be viewed as popular places in the area. We randomly assign each user a home region and 4 other secondary regions. At 8 a.m. every morning, a user leaves his home region in order to visit his secondary regions. When he arrives at a secondary region, it assigns a visit duration between [30,120] min, and moves according to the random waypoint model within the region during this time period. Whenever he starts a new movement (within or towards a region), he selects a speed between [0.33, 2] m/s. After each visit to secondary regions, a user may also visit one of the other 20 regions with probability 0.1, and then goes to the next secondary region. When his visit to the last secondary region ends, he returns to the home region, and repeats this process every day. All simulation settings and default parameters are given in Table I.

[1]There might be some users trying to cheat the task owner by asserting an inflated quality value. These could be avoided using trust building mechanisms [20], but this is out of the scope of the current paper.

TABLE I: Simulation parameters.

Parameter & Settings	Value
Simulation area	1000 m×1000 m
Number of users	100
Number of regions	25
Region size	100 m×100 m
Speed range	[0.33, 2] m/s
Number of secondary regions of each node	4
Visit duration in secondary regions	[30, 120] min
Probability of visiting a random region	0.1
Communication range	30 m
Total simulation time	three weeks
Warm-up period	a week
Number of tasks per user	10
Default expiration period of tasks	3 days
Number of experiments	1000
Default assignment limit (L)	4
Default value of α in Superiority-based protocol	1
Default value of K in Best-K protocol	10

In order to evaluate the performance of our protocols, we let each user generate a sensing task that needs to be performed in a randomly assigned region per day during the next 10 days after the warm-up period and at any time of the day. We compare the proposed task assignment protocols with a spraying-like protocol [22], namely *First-L*, in which task owners assign their tasks to the first L users that they encounter. Although it is highly likely in First-L protocol that the tasks may also be assigned to users who will not be very useful in completing them, it also has some advantages like distributing the tasks in the system earlier and not requiring any pre-calculated utility metrics. We also compare them with Online Task Assignment (NTA) protocol proposed in [11]. In the original design of NTA, tasks are associated with certain workloads (i.e., sensing/processing time). In each encounter with another user u_y, starting from the task with the smallest workload, a task owner u_x greedily assigns his tasks to u_y if the instant processing time of u_y (i.e., as u_x can immediately assign his tasks to u_y) is smaller than the expected processing time of other users in the system. To adapt it to our problem, we use the average time it would take for u_y to go to the task region as the workload of the task since our tasks are location dependent. Besides, as the tasks assumed in our system have negligible sensing times, we do not increase the workload and hence instant/expected processing times of

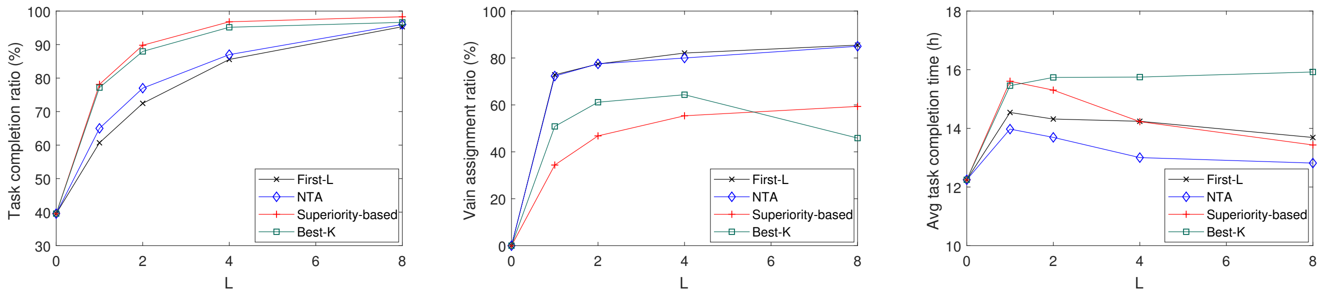

Fig. 3: Comparison of task assignment protocols with different assignment limit (L) values.

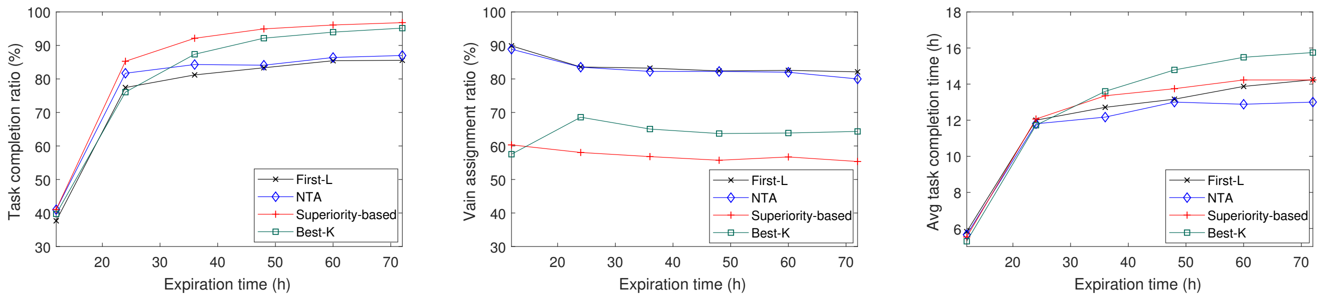

Fig. 4: Comparison of task assignment protocols with different task expiration times.

users after assigning them a task. We also limit the number of assignments per task in NTA to make a fair comparison.

Three performance metrics are used in the evaluations:

- **Task completion ratio**: The proportion of the completed tasks before their expiration time.
- **Vain assignment ratio**: The ratio of the number of assignments which were not profitable for task owners (as they never get the sensed data back or get it after the task is already completed by himself or another user) to the number of all assignments.
- **Average task completion time**: The average time passed since the generation of each task to the moment the task owner obtains the sensed data (unfinished tasks are not included).

B. *Results*

We first compare the performance of the proposed protocols with First-L and NTA protocols. Fig. 3 shows their performance with various assignment limit (L) values. Note that our protocols achieve a significantly higher task completion ratio than the others, and the difference gets as high as 20% when L is small. Moreover, as shown in the middle graph, our protocols perform much better in terms of cost efficiency. That is, they assign the tasks to smaller number of users, but still manage to achieve a larger task completion ratio. On the other hand, First-L protocol has relatively better task completion time per task, but it should be noted that uncompleted tasks are not included here. Thus, it can be said that with First-L protocol, a smaller proportion of the tasks are completed, but they are completed in a shorter time mostly due to uncontrolled, immediate task assignment approach adopted.

Since the relationships between user encounters and regional visits are not considered in NTA, tasks might end up being assigned to users that frequently visit the task region but rarely encounter the task owner, or vice versa. In fact, we observe that when NTA is used as the assignment protocol, a task owner assigns his tasks to the users he encounters most frequently. Because of this, NTA has the smallest average task completion time among all protocols, as once these frequently encountered users perform the task in the task region (if they ever do), they will return the sensed data back to the task owner much more rapidly.

In Fig. 4, we look at the performance of the protocols with different task expiration times. Naturally, a longer expiration period would increase the likelihood of tasks being completed, hence improves the task completion ratio as seen on the left graph. It also results in higher average task completion time due to the same reason. Although the vain assignment ratio does not seem to be much affected by the changes in expiration period of tasks, it is worth noting that the highest vain assignment ratio in all protocols (except Best-K) is when the expiration period is 12 hours because task owners start to assign their task after the task generation, but cannot get the results back as the expiration duration of tasks is too short. Best-K behaves differently, because after the task generation, it takes longer for the task owners to start assigning their tasks since they first need to encounter one of the K users with the highest utility for the task region.

Finally, in Fig. 5 and 6, we analyze the effect of α and K parameters in the performance of Superiority-based and Best-K protocols, respectively. Note that average task completion times for different α and K values are not presented as they do not have a notable difference. In Fig. 5, we observe that the vain assignment ratio steadily decreases with increasing α values, while the task completion ratio begins to decline after $\alpha = 1.5$. This is expected because a large α means that users will be too picky in their task assignments, so they will make fewer assignments, but to users that are highly likely to

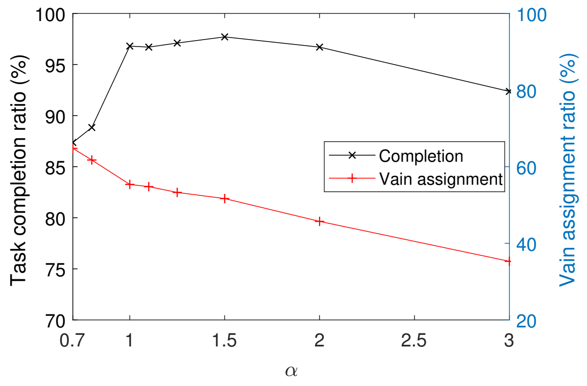

Fig. 5: The impact of α value on the performance of Superiority-based assignment protocol.

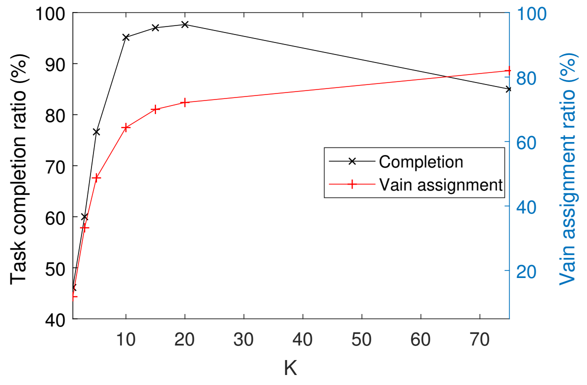

Fig. 6: The impact of K value on the performance of Best-K assignment protocol.

complete the tasks. Therefore, a user who is short of virtual coins might want to use a relatively high α value. Other than that, an α value between 1.5 and 2 should be fine in most cases. In Fig. 6, we see that the vain assignment ratio always gets higher as K increases. In fact, when K is selected too large (75), the performance of Best-K protocol is hardly different than that of First-L protocol, because users start to assign their tasks almost anyone they encounter until the limit is reached.

VI. Conclusion

In this paper, we study the task assignment problem in opportunistic mobile crowdsensing applications. We first derive a metric to measure the utility of users in performing tasks in specific regions and returning the sensed data back to the task requester. Then, we propose two assignment protocols based on the devised metric, and show via simulations that they achieve a high task completion ratio while utilizing user resources efficiently (i.e., less vain assignment ratio).

In our future work, we will examine the assignment of sensing tasks that require consecutive visits to multiple regions (e.g. perform the task at r_2 after getting relevant data from r_1). We will also consider multi-hop based task assignments in which the task could be delegated to another user from the user who is currently assigned to it.

References

[1] W. Gong, B. Zhang, and C. Li, "Task assignment in mobile crowdsensing: Present and future directions," *IEEE network*, vol. 32, no. 4, pp. 100–107, 2018.

[2] Y. Hua, W. He, X. Liu, and D. Feng, "Smarteye: Real-time and efficient cloud image sharing for disaster environments," 04 2015.

[3] F. Yucel and E. Bulut, "Clustered crowd gps for privacy valuing active localization," *IEEE Access*, vol. 6, pp. 23 213–23 221, 2018.

[4] S. He, D. Shin, J. Zhang, and J. Chen, "Near-optimal allocation algorithms for location-dependent tasks in crowdsensing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3392–3405, 2017.

[5] Y. Liu, B. Guo, Y. Wang, W. Wu, Z. Yu, and D. Zhang, "Taskme: Multi-task allocation in mobile crowd sensing," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 403–414.

[6] J. Wang, Y. Wang, D. Zhang, F. Wang, H. Xiong, C. Chen, Q. Lv, and Z. Qiu, "Multi-task allocation in mobile crowd sensing with individual task quality assurance," *IEEE Transactions on Mobile Computing*, vol. 17, no. 9, pp. 2101–2113, 2018.

[7] F. Yucel and E. Bulut, "Joint optimization of system and user oriented task assignment in mobile crowdsensing," in *IEEE Global Communications Conference (GLOBECOM)*, 2019.

[8] M. Xiao, J. Wu, S. Zhang, and J. Yu, "Secret-sharing-based secure user recruitment protocol for mobile crowdsensing," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, 2017, pp. 1–9.

[9] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *IEEE INFOCOM*, 2015, pp. 2254–2262.

[10] Y. Wang, H. Li, and T. Li, "Participant selection for data collection through device-to-device communications in mobile sensing," *Personal and Ubiquitous Computing*, vol. 21, no. 1, pp. 31–41, 2017.

[11] M. Xiao, J. Wu, L. Huang, Y. Wang, and C. Liu, "Multi-task assignment for crowdsensing in mobile social networks," in *IEEE INFOCOM 2015, Kowloon, Hong Kong, April 26 - May 1, 2015*, 2015, pp. 2227–2235.

[12] M. Xiao, J. Wu, L. Huang, R. Cheng, and Y. Wang, "Online task assignment for crowdsensing in predictable mobile social networks," *IEEE Trans. on Mobile Computing*, vol. 16, no. 8, pp. 2306–2320, 2017.

[13] Y. Han and H. Wu, "Minimum-cost crowdsourcing with coverage guarantee in mobile opportunistic d2d networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2806–2818, 2017.

[14] Y. Zhan, Y. Xia, J. Zhang, and Y. Wang, "Incentive mechanism design in mobile opportunistic data collection with time sensitivity," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 246–256, 2018.

[15] Y. Zhan, Y. Xia, Y. Liu, F. Li, and Y. Wang, "Time-sensitive data collection with incentive-aware for mobile opportunistic crowdsensing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 7849–7861, 2017.

[16] J. Phuttharak and S. W. Loke, "Mobile crowdsourcing in peer-to-peer opportunistic networks: Energy usage and response analysis," *Journal of Network and Computer Applications*, vol. 66, pp. 137–150, 2016.

[17] B. Jedari, F. Xia *et al.*, "A survey on routing and data dissemination in opportunistic mobile social networks," *IEEE Communications Surveys and Totorials*, vol. 99, 2013.

[18] I. O. Nunes, C. Celes, I. Nunes, P. O. V. de Melo, and A. A. Loureiro, "Combining spatial and social awareness in d2d opportunistic routing," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 128–135, 2018.

[19] E. Bulut and B. K. Szymanski, "Exploiting friendship relations for efficient routing in mobile social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 12, p. 22542265, 2012.

[20] D. Wu, S. Si, S. Wu, and R. Wang, "Dynamic trust relationships aware data privacy protection in mobile crowd-sensing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2958–2970, 2018.

[21] C. Chen and Z. Chen, "Exploiting contact spatial dependency for opportunistic message forwarding," *IEEE Trans. Mob. Comput.*, vol. 8, no. 10, pp. 1397–1411, 2009.

[22] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Trans. on Networking*, vol. 16, no. 1, pp. 77–90, 2008.