

An Efficient Task Assignment Mechanism for Crowdsensing Systems

Zhuan Shi¹, He Huang^{1,3}, Yu-E Sun^{2,3(✉)}, Xiaocan Wu¹, Fanzhang Li¹,
and Miaomiao Tian³

¹ School of Computer Science and Technology, Soochow University, Suzhou, China

² School of Urban Rail Transportation, Soochow University, Suzhou, China
sunye12@suda.edu.cn

³ Suzhou Institute for Advanced Study,
University of Science and Technology of China, Suzhou, China

Abstract. Crowdsensing has attracted more and more attention in recent years, which can help companies or data demanders to collect large amounts of data efficiently and cheaply. In a crowdsensing system, the sensing tasks are divided into many small sub-tasks that can be easily accomplished by smartphone users, and the companies take advantage of the data collected by all the smartphone users to improve the quality of their services. Efficient task assignment mechanism design is very critical for crowdsensing under some realistic constraints. However, existing studies on task assignment issue are still have many limitations, such as most of them are failed to consider the time budget of smartphone users. Therefore, this work studies the optimal task assignment problem in crowdsensing systems, which can maximize the task completion rate with consideration of the time budget of users. We also prove that the optimal task assignment problem is NP-hard, thus we adopt the linear relaxation and greedy techniques to design a near-optimal crowdsensing task assignment mechanism. We also empirically evaluate our mechanism and show that the proposed task assignment mechanism is efficient.

Keywords: Crowdsensing · Task assignment · Time budget · Approximation algorithm design

1 Introduction

In recent years, with the advent of emerging wireless technologies (*e.g.* 4/5G, Femtocell, Offloading) and micro-embedded sensors (such as digital compass, camera, accelerometer, GPS, proximity sensor, gyroscope, and *etc.*), smartphones and the some other intelligent mobile devices are becoming the crucial central computing devices in our daily lives [2, 12, 14]. Meanwhile, the storage capacities and computing capabilities of the smart devices are getting powerful [10]. Thus, the smartphone not only serves as the common communication and computing device, but it also acts as an information collection and programmable device nowadays. With the increasing number of smartphone users,

crowdsensing (*a.k.a* mobile phone sensing, mobile crowd sensing, participatory sensing) applications have been emerging as a promising paradigm which takes advantage of the smartphones to collect the ubiquitous environment data efficiently and cheaply. Comparing with the traditional static and artificial data collecting methods, the sensed data collected from smartphones also far beyond the scale of traditional ones.

Crowdsensing system borrows from the original idea of crowdsourcing, which is regarded as a distributed problem-solving and business production model. To reduce the production costs, a crowdsourcing system (*e.g.* Amazon Turk) often introduces a large amount of volunteers to solve a complex problem by offering some of them with bonus [7, 21]. Thus, crowdsensing systems can leverage the power of large crowd to complete the large-scale sensing tasks at a lower cost [6]. Considering of the full potential of crowdsensing, many researchers have proposed numerous crowdsensing systems. For instance, literature [17] proposed a crowdsourcing system, named VTrack, for estimating the travel time in urban area to relieve the traffic delay with the built-in GPS. Kumar Rana *et al.* [16] implements a Ear-Phone system for urban noise mapping, which can monitor the environmental noise pollution in urban areas through crowdsourcing data collection.

Many existing studies assume that the volunteers are willing to upload their sensed data to the task demanders without any reward. Unfortunately, due to the limitation of battery and CPU resources, most of the smartphone users are not willing to participate in crowdsensing tasks without payment [8, 20]. Therefore, task allocation issue is playing an important role in mechanism design. There have been large amount of efforts in researching the task allocation problem for the crowdsensing systems [1, 6, 9, 22, 23]. However, most of the existing studies did not consider the time budget of smartphone users. In practice, the smartphone users may not always available during each round of task assignment. Different users may have different time budget for crowdsensing tasks according to their schedules. We can only assign tasks which can be finished by users within their time budgets. Moreover, most of the existing studies restrict that we can only assign no more than one task to each user in each round, which makes the task assignment mechanisms are not very efficient. Only [6] assumes that each user has a time budget, which corresponds to a traveling distance budget. However, the distance budget is not always equal to the time budget. For example, the time budget of one user is 1 h and his moving speed is 5 km/h, which does not mean that this user is willing to achieve the tasks within 5 km. In a more realistic scenario, most of the users only want to achieve the tasks nearby. To ensure this, users in [6] need to set their distance budgets to small values, which will decrease the performance of the task assignment mechanism. To deal with the above challenges, more practical and efficient crowdsensing task assignment mechanisms are needed.

This work is focused on an efficient task assignment mechanism design for mobile sensing systems, which can maximize the task completion rate with consideration of time budget of the smartphone users. We define the task completion

rate as the ratio of the number of tasks that are assigned to users and the total number of tasks supplied by platform. In general, the more tasks are finished by users, the more improvement of the quality of service supplied by the platform we can get. Thus, to design an efficient task assignment mechanism that maximizes the task completion rate can help to improve the service performance. To make our model more efficient, we also allow the platform assign multiple tasks to users in each round of assignment. However, it is a hard work to design an efficient mechanism under our crowdsensing system model. We can prove that the optimal task assignment problem studied in this work is NP-hard. To tackle this NP-hardness, we use LP (*a.k.a* linear programming) relaxation and greedy techniques to design a near-optimal task assignment algorithm. The main contributions of this work can be summarized as follows:

- To the best of our knowledge, we are the first to study crowdsensing task assignment mechanism which assumes that each user has a time budget and aims at maximizing the task completion rate.
- We design a near-optimal task assignment algorithm, which based on LP relaxation and greedy techniques.

2 Problem Model

In this section, we will introduce the crowdsensing system model and the optimal task assignment problem we studied in this paper.

2.1 Crowdsensing System Model

We study a mobile sensing system, which is composed of a mobile sensing platform and a set of smartphone users. In our model, the platform will assign the sensing tasks to the smartphone users, and the users will be paid accordingly after finishing their tasks. For instance, if some application providers are willing to draw the noise map of a city, the sensing platform will assign the tasks of collecting noise to the smartphone users. Each selected user need to collect noise data from a certain number of monitoring points and submit the sensing data to the platform. In this work, we assume that each task can only be assigned to one user, and leave the case that one task can be accomplished by multiple users as our further work.

Our task assignment mechanism runs periodically. In each round, we assume that there exist m tasks in the platform, which are in the set $T = \{t_1, t_2, t_3, \dots, t_m\}$. Each task $t_j (j \in [1, m])$ can be denoted as $t_j = \{L_j, d_j\}$, where L_j is the location of t_j and d_j is the detailed description of t_j . After reading the description of tasks, there are n smartphone users want to join in the tasks. We use $U = \{u_1, u_2, u_3, \dots, u_n\}$ to denote the set of smartphone users. Each user $u_i \in U$ will submit the sensing plan to the platform. The sensing plan of each u_i can be described as $s_i = \{L_i, r_i, time_budget_i, b_i\}$, where L_i is the geographical location of u_i , r_i is the acting radius of u_i , $time_budget_i$ is the

upper bound of u_i 's working time duration, and b_i is the per-unit time bid of u_i for completing the tasks. Usually, people only want to do the tasks nearby. Thus, we assume that each user only wants to do the tasks that are located in the circle within the geographical location L_i as the center and r_i as the radius. Moreover, we also consider the time budget of each user, which is not considered in most of the previous studies. Thus, the total working time of u_i is defined as the time consuming for achieving all the tasks that assigned to u_i , which should not exceed $time_budget_i$.

2.2 The Optimal Task Assignment Problem

In general, the more tasks are accomplished, the better performance of the crowdsensing system can get. Therefore, in this work, maximizing the accomplished tasks is the optimal object during the task assignment procedure. The so-called optimal task assignment problem is aiming at maximizing the number of the tasks accomplished, which can be represented as $\max \sum_{t_j \in T} x_j$. Here, $x_j = 1$ denotes that t_j is assigned to user, and $x_j = 0$ means that t_j is not assigned to any smartphone user. To assign tasks to users optimally, we first introduce a variable $c_{i,j}$ to indicate whether t_j is within u_i 's activity range. If t_j is within u_i 's activity range, t_j can be assigned to u_i and we set $c_{i,j} = 1$; otherwise, t_j cannot be assigned to u_i and we set $c_{i,j} = 0$. Further, we use a binary variable $x_{i,j} = \{0, 1\}$ to denote whether t_j is assigned to user u_i or not. According to the constraints mentioned above, we can conclude that $x_{i,j}$ should be no more than $c_{i,j}$ (i.e. $x_{i,j} \leq c_{i,j}$). From the relationship between x_j and $x_{i,j}$, it's easy to get that $x_j = \sum_{u_i \in U} x_{i,j}$.

Then, we use $time_{i,j}$ to denote the consumption time that u_i requires to achieve the task t_j . Note that $time_{i,j}$ includes the time of u_i arrives to the required location of t_j from L_i , the required time to accomplish t_j , and the time return to L_i from t_j . In our model, we assume that each user has a time budget. Thus, we will ensure that all the tasks that assigned to u_i satisfies $\sum_{t_j \in T} x_{i,j} time_{i,j} \leq time_budget_i$ in the proposed task assignment mechanism. Since $\sum_{t_j \in T} x_{i,j} time_{i,j}$ is the upper bound of time for u_i to finish all the tasks that assigned to it, the time budget of each user will not be violated.

Finally, we use p_i to represent the payment of u_i after the smartphone user i accomplishes all the assigned tasks. In this work, we assume that $p_i = \sum_{t_j \in T} x_{i,j} time_{i,j}$. We further assume that the platform has a general budget constraint B in terms of the task assignment plan, so the total payment of users can not exceed B , which should satisfy the constraint $\sum_{u_i \in U} p_i \leq B$.

According to the analysis above, the optimal task assignment problem studied in this work can be described as an integer problem (IP(1)):

$$\max \quad g(x) = \sum_{t_j \in T} x_j,$$

subject to

$$\begin{cases} x_{i,j} \leq c_{i,j}, \forall i, \forall j \\ x_j = \sum_{u_i \in U} x_{i,j}, \forall j \\ \sum_{t_j \in T} x_{i,j} \text{time}_{i,j} \leq \text{time_budget}_i, \forall i \\ \sum_{u_i \in U} (b_i \cdot \sum_{t_j \in T} x_{i,j} \text{time}_{i,j}) \leq B \\ x_j = \{0, 1\}, \forall j \\ x_{i,j} = \{0, 1\}, \forall i, \forall j \end{cases}$$

We can prove that the proposed IP(1) is NP-hard.

Theorem 1: The optimal task assignment problem studied in this paper is NP-hard.

Proof: Consider a simple situation where there is only one user. In this case, the problem we studied can be viewed as a 0–1 knapsack problem, where the time that the user consumes to accomplish tasks can be viewed as the weight of goods in 0–1 knapsack problem, the volume of the knapsack is the time budget of this user and the value of goods in 0–1 knapsack problem are all equal to one unit in our problem. Since the 0–1 knapsack problem is a typical NP-hard problem, the optimal task assignment problem studied in this paper is NP-hard.

As is known to all, there exists a PTAS (Polynomial-time approximation scheme) mechanism based on dynamic programming to solve the 0–1 knapsack problem. However, this mechanism can only be used in the case that there is one user. For the case that there exist multiple users, we need to design other mechanism to solve the NP-hard problem we studied in this paper.

3 Approximation Task Assignment Mechanism

To tackle this NP-hardness we studied, we adopt the linear programming (LP) relaxation technique to design a near optimal task assignment mechanism in this section. In the following, we will give the details of the proposed mechanism.

3.1 The Relaxation of IP(1)

We first relax IP(1) to linear programming LP(2) by replacing $x_{i,j} = \{0, 1\}$ with $0 \leq x_{i,j} \leq 1$, and replace $x_j = \{0, 1\}$ with $0 \leq x_j \leq 1$. Then, the optimal task assignment problem is reformulated as the following relaxed LP problem (LP(2)):

$$\max \quad g(x) = \sum_{t_j \in T} x_j,$$

subject to

$$\begin{cases} x_{i,j} \leq c_{i,j}, \forall i, \forall j \\ x_j = \sum_{u_i \in U} x_{i,j}, \forall j \\ \sum_{t_j \in T} x_{i,j} \text{time}_{i,j} \leq \text{time_budget}_i, \forall i \\ \sum_{u_i \in U} (b_i \cdot \sum_{t_j \in T} x_{i,j} \text{time}_{i,j}) \leq B \\ 0 \leq x_j \leq 1, \forall j \\ 0 \leq x_{i,j} \leq 1, \forall i, \forall j \end{cases}$$

The relaxed problem LP(2) has a polynomial number of variables and constraints, thus it can be solved optimally in polynomial time. Assume that the optimal solution of LP(2) is O_{LP} , the optimal solution of IP(1) is O_{IP} and F_{IP} is a feasible solution of IP(1). Obviously, we can get O_{LP} in polynomial time by solving LP(2) optimally. However, either $x_{i,j}$ or x_j should be equal to 0 or 1 in the task assignment model, which means a value between 0 to 1 is meaningless. Thus, O_{LP} is often not a feasible solution of IP(1). To solve this problem, we need to design a mechanism to convert the O_{LP} to a feasible solution of IP(1).

3.2 A Near-Optimal Solution for the Optimal Task Assignment Problem

To convert O_{LP} to a feasible solution of IP(1), we need to select the allocation relationships which satisfy $0 < x_{i,j} < 1$ in O_{LP} , and set them equal to 1. In this work, we propose a greedy-like mechanism, which is described in Algorithm 1.

During the process of assigning the tasks, the platform applies Algorithm 1. Before applying it, we have already had the optimal solution of linear programming O_{LP} , the set of users sensing plans S , the set of tasks T released by the platform, the budget constraint B of the platform. First, we introduce a variable B' , whose original value is $B' = 0$, to represent the actual payment of the platform. We introduce a variable t'_i for each user u_i , whose original value $t'_i = 0$, to represent u_i 's actual working time. At the beginning, we sort all the $x_{i,j} \in O_{LP}$ in descending order. When $x_{i,j}$ was scanned, we decide first whether t_j has been assigned to users, then whether the actual working time that u_i consumed is within the upper time limit time_budget_i if we assign t_j to u_i , and finally whether the actual payment of the platform is no more than the budget constraint B if we assign t_j to u_i . If the above three conditions are all met at the same time, we will assign t_j to u_i , which means sets $x_{i,j}^f = 1$ and updates the value of B' and t'_i . Otherwise, we set $x_{i,j}^f = 0$. Then we decide the next one $x_{i,j}$ in the ordered list. We repeat the process again and again until all the $x_{i,j}^f$ has been decided. At last, we output the set $F_{IP} = \{x_{i,j}^f\}_{u_i \in U, t_j \in T}$. The value correspondent to the set F_{IP} is the final result of our winner determination mechanism.

Algorithm 1. Approximation task assignment mechanism

Input: the optimal solution of LP(2): O_{LP} , the sensing plan of users $S = \{s_i\}_{i \leq n}$, the set of tasks T , the budget B ;

Output: the feasible solution of IP(1): F_{IP} ;

```

1: Set  $B' = 0$ ;
2: for each  $t_i \in T$  do
3:   Set  $x'_j = 0$ ;
4:   for each  $s_i \in S$  do
5:     Set  $time'_i = 0$ ;
6:   Sort all the  $x_{i,j} \in O_{LP}$  in descending order;
7:   for each  $x_{i,j}$  in the sorted list do
8:     if  $x'_j < 1$  then
9:       if  $time'_i + time_{i,j} \leq time\_budget_i$  then
10:        if  $B' + b_i time_{i,j} \leq B$  then
11:          Set  $x^f_{i,j} = 1$ ;
12:          Set  $x'_j = 1$ ;
13:          Set  $B' = B' + b_i * time_{i,j}$ ;
14:          Set  $time'_i = time'_i + time_{i,j}$ ;
15:       if  $x_{i,j} < 1$  then
16:         Set  $x^f_{i,j} = 0$ ;
17: return  $F_{IP} = \{x^f_{i,j}\}_{u_i \in U, t_j \in T}$ ;
```

4 Simulation Results and Analysis

In this section, we will provide the results of simulation experiment of the proposed algorithm, thus to analyze and verify its practical performances. Since the target of our mechanism is to maximize the task completion rate, we will give the formal definition first.

Task completion rate: We define the task completion rate as the ratio between the number of tasks that assigned to users and the total number of tasks that supplied by the platform (*i.e.* $\eta = \sum_{t_j \in T} x_j / n$).

In the simulation, we assume that the tasks and users are located in a $200 * 200$ area. In our model, we assume that $time_{i,j}$ includes the time of u_i reaches t_j from L_i , to accomplish t_j , and return to L_i from t_j . Thus, we set that the completion time that u_i accomplish t_j is (which does not include the time that users consume to reach t_j and return to L_i from t_j) uniformly distributed in the range of $[1/3, 1/2]$, and the moving speed of users are uniformly distributed in the range of $[40, 50]$. Suppose for each user in the set, the working time, radius and per unit time bid are uniformly distributed in the range of $[6, 8]$, $[25, 35]$ and $[35, 50]$ respectively. Finally, all the results are the average value of data from 2000 independent experiments. Meanwhile, in order to testify the stability of the algorithm, we respectively conduct our experiments in uniform distribution and hot spot distribution according to the distribution of the users.

Uniform distribution model: In this model, all the geographical location of tasks and users are uniformly distributed in a $200 * 200$ area.

Hot spot distribution model: In this model, the geographical location of tasks are uniformly distributed in a $200 * 200$ area, the geographical location of 90% users are uniformly distributed in a $16 * 16$ square hot spot area, and 10% users are located uniformly in other area.

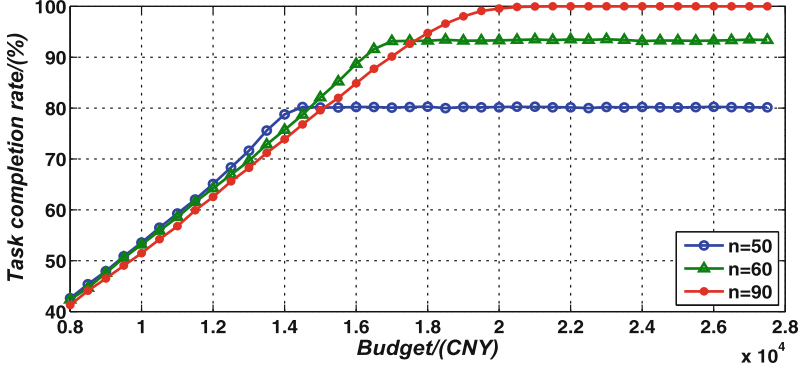


Fig. 1. The relationship between the task completion rate and the budget constraint under uniform distribution

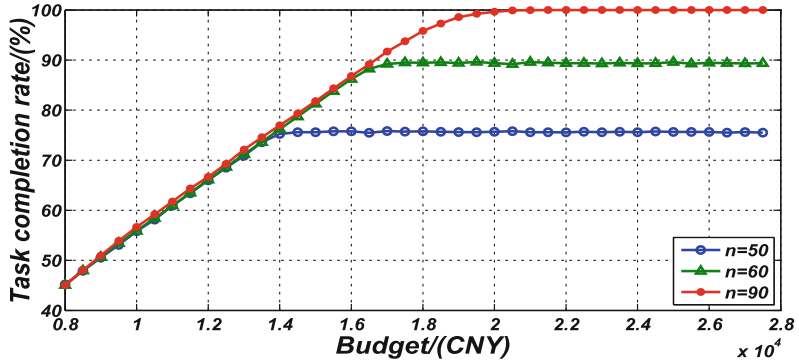


Fig. 2. The relationship between the task completion rate and the budget constraint under hot spot distribution

Figures 1 and 2 show the relationship between task completion rates and the budget constraint of the platform when the number of users is 50, 60, and 90 in uniform and hot spot distribution. Apparently, with the budget constraint increasing, the task completion rate first increase and then level off. This is mainly because the platform has no enough money to pay the users who are willing to do the tasks when the budget constraint is relatively low. Thus, larger

budget means the platform can assign more tasks to users in this case, the number of the accomplished tasks increase with the increasing of the budget constraint. Due to each user has a time budget, the platform only can assign limit tasks to each user. Thus, the task completion rate will not increase with the budget constraint when there are no user can be assign more tasks, and the task completion rate curve will level off. Obviously, more user can do more tasks when the budget is larger enough, thus the task completion rates should increase with the increasing number of the users, which is confirmed by Figs. 1 and 2.

5 Related Work

In recent years, many state-of-art studies on crowdsensing systems have been proposed [3, 4, 11, 15–19]. For example, [16] and [17] are making effort on road traffic monitoring. In [4], the authors proposed a crowdsensing based street holes detection system by letting users share vibration and location information captured by smartphones.

Task assignment mechanism is critical in crowdsensing system design, which determines whether a crowdsensing system can achieve good service quality. Recent years, the optimal crowdsensing task assignment problem has been widely studied and many efficient mechanisms have been proposed. For instance, Zhao *et al.* designed an online mechanism for tasks allocation with consideration of budget constraint [22]. In [5], Feng *et al.* proposed two allocation mechanisms for mobile crowdsensing system in two different scenarios with consideration of uncertain arrivals of tasks, strategic behaviors. In [23], Zhao *et al.* proposed two fair energy-efficient approximation algorithms for allocating tasks optimally in crowdsensing systems. In [13], an allocation mechanism based on all-pay auctions is studied, which aims at attracting more users' participation while maximizing the organizer's profit. In [1], the authors proposed an online model to efficiently decide the most appropriate set of users to achieve the incoming tasks. Many tasks of the crowdsensing systems are location dependent, thus the users need to spend a certain amount of time to travel around when they finish the tasks. In practice, many users only have limited times to achieve the tasks. However, the time budget of users is only considered in [6]. Even in [6], the authors viewed the time budget of users as distance budget, which is not very reasonable. Thus, we will solve this challenge in this work and propose an efficient task assignment mechanism.

6 Conclusion

In this paper, we propose a near-optimal crowdsensing task assignment mechanism, that consider the time budget of users, support assigning multi-tasks to each user in each round of assignment and maximizes the task completion rate. Compared with the existing studies, the crowdsensing system model studied in this paper is more practical and efficient. The detailed experimental results show that our task assignment mechanism has good performance.

Acknowledgements. This work is partially supported by National Natural Science Foundation of China (NSFC) under Grant No. 61572342, No. 61303206, No. 61672369, Natural Science Foundation of Jiangsu Province under Grant No. BK20151240 and No. BK20161258, China Postdoctoral Science Foundation under Grant No. 2015M580470 and No. 2016M591920. Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAEET). Any opinions, findings, conclusions, or recommendations expressed in this paper are those of author(s) and do not necessarily reflect the views of the funding agencies (NSFC).

References

1. Boutsis, I., Kalogeraki, V.: On task assignment for real-time reliable crowdsourcing. In: IEEE ICDCS 2014, pp. 1–10 (2014)
2. Chatzimilioudis, G., Konstantinidis, A., Laoudias, C., Zeinalipour-Yazti, D.: Crowdsourcing with smartphones. *IEEE Internet Comput.* **16**(5), 36–44 (2012)
3. Chon, Y., Lane, N.D., Li, F., Cha, H., Zhao, F.: Automatically characterizing places with opportunistic crowdsensing using smartphones. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing (Ubicomp 2012), pp. 481–490 (2012)
4. Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S., Balakrishnan, H.: The pothole patrol: using a mobile sensor network for road surface monitoring. In: ACM MobiSys 2008, pp. 29–39 (2008)
5. Feng, Z., Zhu, Y., Zhang, Q., Zhu, H., Yu, J., Cao, J., Ni, L.M.: Towards truthful mechanisms for mobile crowdsourcing with dynamic smartphones. In: IEEE ICDCS 2014, pp. 11–20 (2014)
6. He, S., Shin, D.-H., Zhang, J., Chen, J.: Toward optimal allocation of location dependent tasks in crowdsensing. In: IEEE INFOCOM 2014, pp. 745–753 (2014)
7. Howe, J.: Crowdsourcing: How the Power of the Crowd is Driving the Future of Business. Random House, New York (2008)
8. Huang, H., Sun, Y.-E., Li, X.-Y., Chen, S., Xiao, M., Huang, L.: Truthful auction mechanisms with performance guarantee in secondary spectrum markets. *IEEE Trans. Mob. Comput.* **14**(6), 1315–1329 (2015)
9. Jin, H., Su, L., Chen, D., Nahrstedt, K., Xu, J.: Quality of information aware incentive mechanisms for mobile crowd sensing systems. In: ACM MobiHoc 2015, pp. 167–176 (2015)
10. Kanhere, S.S.: Participatory sensing: crowdsourcing data from mobile smartphones in urban spaces. In: Hota, C., Srimani, P.K. (eds.) ICDCIT 2013. LNCS, vol. 7753, pp. 19–26. Springer, Heidelberg (2013). doi:[10.1007/978-3-642-36071-8_2](https://doi.org/10.1007/978-3-642-36071-8_2)
11. Koukoulidis, E., Peh, L.-S., Martonosi, M.R.: Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory. In: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys 2011), pp. 127–140 (2011)
12. Lane, N.D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., Campbell, A.T.: A survey of mobile phone sensing. *IEEE Commun. Mag.* **48**(9), 140–150 (2010)
13. Luo, T., Tan, H.-P., Xia, L.: Profit-maximizing incentive for participatory sensing. In: IEEE INFOCOM 2014, pp. 127–135 (2014)

14. Ma, T., Zhou, J., Tang, M., Tian, Y., Al-Dhelaan, A., Al-Rodhaan, M., Lee, S.: Social network and tag sources based augmenting collaborative recommender system. *IEICE Trans. Inf. Syst.* **98**(4), 902–910 (2015)
15. Rai, A., Chintalapudi, K., Padmanabhan, V.N., Sen, R.: Zee: zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom 2012)*, pp. 293–304 (2012)
16. Rana, R.K., Chou, C.T., Kanhere, S.S., Bulusu, N., Hu, W.: Ear-phone: an end-to-end participatory urban noise mapping system. In: *ACM/IEEE IPSN 2010*, pp. 105–116 (2010)
17. Thiagarajan, A., Ravindranath, L., LaCurts, K., Madden, S., Balakrishnan, H., Toledo, S., Eriksson, J.: Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In: *ACM Sensys 2009*, pp. 85–98 (2009)
18. Xu, W., Huang, H., Sun, Y.-E., Li, F., Zhu, Y., Zhang, S.: DATA: a double auction based task assignment mechanism in crowdsourcing systems. In: *8th International ICST Conference on Communications and Networking in China (CHINACOM 2013)*, pp. 172–177 (2013)
19. Yan, T., Kumar, V., Ganesan, D.: Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones. In: *MobiSys 2010*, pp. 77–90 (2010)
20. Yang, D., Xue, G., Fang, X., Tang, J.: Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing. In: *ACM Mobicom 2012*, pp. 173–184 (2012)
21. Yuen, M.-C., King, I., Leung, K.-S.: A survey of crowdsourcing systems. In: *IEEE Third International Conference on Privacy, Security, Risk and Trust (PASSAT 2011)* and *IEEE Third International Conference on Social Computing (SocialCom 2011)*, pp. 766–773 (2011)
22. Zhao, D., Li, X.-Y., Ma, H.: How to crowdsource tasks truthfully without sacrificing utility: online incentive mechanisms with budget constraint. In: *IEEE INFOCOM 2014*, pp. 1213–1221 (2014)
23. Zhao, Q., Zhu, Y., Zhu, H., Cao, J., Xue, G., Li, B.: Fair energy-efficient sensing task allocation in participatory sensing with smartphones. In: *IEEE INFOCOM 2014*, pp. 1366–1374 (2014)

Cloud Computing and Security

Second International Conference, ICCCS 2016, Nanjing,
China, July 29-31, 2016, Revised Selected Papers, Part II

Sun, X.; Liu, A.X.; Chao, H.-C.; Bertino, E. (Eds.)

2016, XX, 636 p. 261 illus., Softcover

ISBN: 978-3-319-48673-4