

Quality-Assure and Budget-Aware Task Assignment for Spatial Crowdsourcing

Qing Wang¹, Wei He^{1(✉)}, Xinjun Wang^{1,2}, and Lizhen Cui¹

¹ School of Computer Science and Technology, Shandong University, Jinan, China
wang_qing@mail.sdu.edu.cn, {hewei, wxj, clz}@sdu.edu.cn

² Dareway Software Co., Ltd, Jinan, China

Abstract. With the increasingly ubiquity of mobile devices and the rapid development of communication technologies, spatial crowdsourcing has become a hot topic research among academic and industry community. As participants may possess different capabilities and reliabilities, as well as the changeable locations and available time slots of both tasks and potential workers, a major challenge is how to assign spatial tasks to appropriate workers from lots of potential applicants, which should assure the result quality of the crowdsourcing task. Also, as different workers may receive variable rewards for the same task, the crowdsourcing budget renders task assignment more complicated. This paper focuses on the issue of quality assurance for task assignment in spatial crowdsourcing while considering budget limitation. The problem is first modeled as Quality-assure and Budget-aware Task Assignment (QBTA) problem. Then two two-phase greedy algorithms are proposed. Finally, experiments are conducted to show the effectiveness and efficiency of the algorithms.

Keywords: Spatial crowdsourcing · Task assignment · Quality assurance

1 Introduction

With the increasingly ubiquity of mobile devices [1] and the rapid development of ultra-broadband wireless networks (e.g., 4G) [2], spatial crowdsourcing has attracted more and more attention in both the academic and industry community. It aims to allocate each spatial task (i.e., the task related to its location and time) to a set of workers, whom are required to reach the predetermined physical position to perform the task (e.g., taking a picture or checking in a spot) before the expiration time. In order to attract more participants, each worker who has been assigned and performed the task is probably rewarded in any form of incentive such as monetary payment or diverse coupons which can be exchanged for goods or services [3].

As the potential workers may possess different capabilities and reliabilities, a major impediment to the success of any spatial crowdsourcing [1] is how to guarantee the quality of results (e.g., the accuracy of the collected results) obtained from the workers [3]. An intuitive idea of quality assurance for spatial crowdsourcing is assigning tasks to workers with high reputations and abilities. Kazemi et al. defined a confidence level for every spatial task and stated that each assignment is accepted only if the confidence

of candidate workers is higher than a certain threshold [1]. In their approach, a parameter “maxT” is introduced for each worker to limit the maximum number of tasks assigned to him/her in order to avoid global imbalance of task assignment. As a result, each task will be assigned to as few workers as possible which satisfies the confidence level of the task. However, it is not always reasonable because the task publisher may expect a large number of different workers to perform the same task, rather than a small set of skilled workers. For example, in order to check the condition of the billboards at different locations, an advertising agent expects to collect a great many photos of the billboards to assure a right decision could be made on whether the billboards need to be repaired/replaced or not. Note that the photos with diverse directions and different time of photography always show different conditions. Therefore, to obtain the full view of the billboards, a straightforward method is to employ a large number of workers to perform the task, which provides a high probability to get more satisfied and diverse results. But it is usually impractical because the task publisher will provide a personal reward for every worker who performs the task, which may be resulted in an expensive cost on workers’ rewards for the task publisher. Han Yu et al. aim to figure out a set of workers for each spatial task with the maximum expected quality of results and the total rewards within a pre-specified budget [3]. However, it is not reasonable for the assumption that every worker will perform the assigned task correctly. As the potential workers may locate at various positions to perform the task, which probably leads to different traffic costs (i.e., time, distance and money), the expensive traffic costs may discourage the workers to reach the task location and perform the task correctly before expiration time in case that the traffic costs exceed their expectations.

In this paper, we focus on the issue of quality assurance (i.e., at least a given number of workers could reach the task location and perform the task correctly before the expiration time) for task assignment in spatial crowdsourcing while considering a total the budget limitation. For one specific spatial task, the objective is to figure out a set of workers from large numbers of potential applicants to perform the task, which satisfies both the pre-specified confidence requirement and budget limitation. The problem is first modeled as Quality-assure and Budget-aware Task Assignment (QBTA) problem. A confidence score (CS) is proposed for each individual worker to indicate the probability that he/she will reach the task location and correctly perform the specific spatial task before the expiration time. Based on the confidence score of individual workers, the Aggregate Confidence Score (ACS) is computed for a worker set to indicate the total confidence for the specific spatial task, which is the probability that at least a given number of workers in the worker set will reach the task location and perform the task correctly before the expiration time. Then, we propose a two-phase greedy algorithm (Quality_Sensitive) to figure out a worker set with higher quality for QBTA problem. Additionally, based on the Quality_Sensitive, another greedy algorithm (Cost_Sensitive) is developed to find out a worker set with less total rewards for QBTA problem. Finally, experiments on the real-world dataset are conducted to show the effectiveness and efficiency of our algorithms.

The rest of the paper is organized as follows. In Sect. 2, we review the related work and then define our problem formally in Sect. 3. We present our approaches namely

Quality_Sensitive and Cost_Sensitive in Sect. 4. Section 5 reports the results of our experiments. Finally, we conclude the paper in Sect. 6.

2 Related Work

With the ubiquity of mobile devices, spatial crowdsourcing is emerging as a new platform, enabling spatial tasks assigned to and performed by human workers [4]. Based on the motivation of the workers, spatial crowdsourcing has been classified into two classes: reward-based and self-incentivised [4]. With reward-based spatial crowdsourcing [6], every spatial task has a price and workers will receive a certain reward for every spatial task they perform correctly. The self-incentivised spatial crowdsourcing is for workers who are self-incentivised to perform tasks voluntarily [4]. We focus on the former class of spatial crowdsourcing.

As the fact that multiple available workers may compete for the same spatial task, the research on finding efficient solutions of task assignment in spatial crowdsourcing has attracted significant interest. Shahabi et al. focus on how to assign every worker his/her nearby tasks with the objective of maximizing the overall number of assigned tasks [4]. Deng et al. aim to find a task schedule for a worker that maximizes the number of performed tasks [2]. In [7], Yu Li et al. recommend an optimal route for a crowdsourcing worker with the maximum rewards of tasks along the route. Note that the workers with different levels of competence [3] may provide the results with different qualities and may receive different rewards for the same task. A major challenge for the reward-based spatial crowdsourcing is how to find an efficient solution of task assignment satisfying both the quality requirement and budget limitation of the task. Han Yu et al. determine the reward of a worker based on his/her past performance [8–10] and propose a *Budget-TASC* algorithm to find a task allocation plan to maximize the expected quality of the result while satisfying the budget limitations [3]. Yuko Sakurai et al. also propose a method for constructing an appropriate set of reward plans under a requester's constraints on budget and required accuracy [9]. In [1], a *HGR* algorithm is proposed to assign the task whose result is in the form of data modality, such as a binary value (0/1) to a set of workers. In reality, spatial tasks are required to support different other modalities of results (e.g., text, photos). Therefore, the algorithms aim to assign the task whose result is in more complex form of modality are needed.

3 Problem Definitions and Formulation

In this paper, we focus on how to find an efficient solution of task assignment in spatial crowdsourcing. As the fact that the spatial task always needs to be performed correctly by multiple workers in order to assure the quality of task result and the task publisher always offers variable rewards for different workers based on their track records and ratings, the objective of task assignment is to figure out a set of workers from large numbers of potential workers for each spatial task, which assures: (i) at least a given number of workers could reach the task location and correctly perform the task before expiration time, and (ii) the total rewards of the selected workers would not exceed the

budget. In this section, we first formally define the QBTA problem and then give an example to illustrate the problem. Last, we discuss the confidence evaluation model.

3.1 Problem Definitions

Definition 1 (Worker). A worker w is represented as a 4-tuple $\langle r, rk, l_w, t_w \rangle$, which denotes that the worker w with a reputation r and a rank rk locates in the location l_w at time t_w . The reputation r indicates the worker's credibility and computed based on his/her historical ratings and track records, which is used to measure the probability that he/she will perform the task and submit a satisfied result if he/she has been assigned the task. For simplicity of discussions, workers are classified as having *high* (H), *medium* (M) or *low* (L) ranks based on their reputations [3], i.e., $rk = H/M/L$.

In many crowdsourcing applications, such as *Uber* [11], the worker will receive a rating from the task publisher to evaluate the result quality after he/she has performed the task. Many approaches [12–15] can be used to calculate the worker's reputation based on his/her historical ratings and track records. In this paper, the reputation $r(w)$ of worker w is calculated as follows:

$$r(w) = \frac{EX(R)}{M} \text{ and } EX(R) = \frac{1}{n} \sum_{i=1}^n R_i. \quad (1)$$

Where n is the total number of tasks assigned to w , $R_i (1 \leq i \leq n)$ is the rating of task t_i the worker w received, M is the maximum rating the task publisher could give.

Definition 2 (Spatial Task). A spatial task t is represented as a 5-tuple $\langle l_t, et, minW, rew, B \rangle$, which denotes that the task t should be performed in the location l_t before the expiration time et by at least $minW$ number of workers correctly. rew is a 2-tuple $\langle p_H, p_M \rangle$ to state the rewards for workers with *high* and *medium* ranks, respectively. Similar to [3], we assume that a task publisher would not to engage a *low-rank* worker. $B(t)$ is the budget limitation of the task t , which means that t should be assigned to a set of workers which the total rewards are less than $B(t)$.

Definition 3 (Confidence Score CS). Confidence Score $CS(t, w)$ is a probability that the worker w will reach the task location and correctly perform the task t before the expiration time, which indicates the confidence level of the worker w for task t .

Definition 4 (Aggregate Confidence Score ACS). Aggregate Confidence Score $ACS(t, W)$ is the probability that at least $minW_t$ number of workers in a worker set W will reach the task location and perform the task t correctly before the expiration time, which indicates the confidence level of the worker set W for the task t .

Definition 5 (Confidence Requirement CR). The Confidence Requirement of task t is represented as $CR_t(conf_t, minW_t)$, which states that t should be assigned to a set of workers W , which satisfies that $ACS(t, W) \geq conf_t$ and $|W| \geq minW_t$.

Based on the definitions above, the issue of task assignment for a spatial crowd-sourcing task is formalized as the QBTA problem, which described as follows:

Given a spatial tasks t and a set of workers W , the objective is to figure out a subset $W^*(t)$ of W which satisfies $CR_t(conf_t, \min W_t)$ and $B(t)$ of the task t . The QBTA problem can be expressed as follows:

$$\begin{aligned}
 & \text{subject to: } ACS(t, W^*(t)) \geq conf_t, |W^*(t)| \geq \min W_t \\
 & \sum_{w \in W^*(t)} rew(w) \leq B(t), W^*(t) \text{ is the subset of } W.
 \end{aligned} \tag{2}$$

3.2 An Example of QBTA Problem

An example of QBTA problem is shown in Fig. 1, where there is one spatial task and ten potential workers. The task t is located in position (5, 5), which will expire after 10 time units and is expected to be performed by at least 3 workers correctly. The task publisher would offer 6-cent and 4-cent rewards for each worker with a high rank and medium rank, respectively and spend at most 25 cents on workers' rewards totally (i.e., $B(t) = 25$). Also, the task publisher requires that the probability of at least 3 workers correctly perform the task is more than 0.6 (i.e., $CR_t(0.6, 3)$). Based on the track records and ratings of each worker w_i , $r(w_i)$ is computed using Eq. (1) and then his/her rank $rk(w_i)$ and reward $rew(w_i)$ are determined. Then, the $CS(t, w_i)$ will be computed by the following confidence evaluation model in the next subsection. Thereafter, our objective is to figure out a set of workers $W^*(t)$ from the ten workers for task t which satisfies $CR_t(0.6, 3)$ and $B(t)$. In this example, task t could be assigned to a worker set $\{w_3, w_4, w_6, w_8\}$ or $\{w_3, w_4, w_6, w_9\}$ which satisfies $CR_t(0.6, 3)$ and $B(t)$.

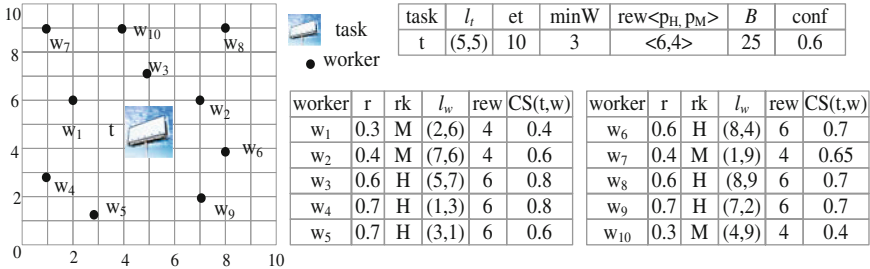


Fig. 1. An example of QBTA problem

3.3 Confidence Evaluation Model

We propose a confidence evaluation model to calculate the confidence score of an individual worker and the aggregate confidence score of a worker set.

Given the spatial task $t(l_t, et, \min W, B)$ and a set of workers $W\{w_1, w_2, \dots, w_m\}$, for $1 \leq i \leq m$, $CS(t, w_i)$ depends on three main factors: (1) worker's reputation $r(w_i)$, a worker with a higher reputation generally provides more reliable results than workers

with lower reputations [3]. (2) Worker's traffic distance to the task location $dis(w, t)$. A further traffic distance always leads to an expensive traffic cost, which renders the worker unwilling to reach the task location. (3) Worker's available time $\Delta t(t, w_i)$ to perform the task after reaching the task location, which is calculated by Eq. (3). The abundant $\Delta t(t, w_i)$ always promotes the worker w_i to offer a more satisfied result.

$$\Delta t(t, w_i) = et(t) - t_{w_i} - \tau(w_i, t) \text{ and } \tau(w_i, t) = \frac{dis(w_i, t)}{speed_{w_i}}. \quad (3)$$

Where $\tau(w, t)$ and $dis(w, t)$ denote the traffic time and distance from l_w to l_t , respectively $speed_w$ is the (constant) traffic speed of worker w . Note that any distance function can be used to calculate $dis(w, t)$ only if it satisfies the triangle inequality, such as Euclidean distance, Manhattan distance and road network distance [7].

In a word, $CS(t, w_i)$ is calculated as follows:

$$CS(t, w_i) = r(w_i) * \delta(l_{w_i}, l_t) * f(w_i, t). \quad (4)$$

Where $\delta(l_{w_i}, l_t)$ is a function calculating the discount to the worker's reputation as a result of his/her location proximity to the task location [3]. It is defined as:

$$\delta(l_{w_i}, l_t) = 1 - \max[0, \min[\log_D(d(l_{w_i}, l_t)), 1]] \quad (5)$$

where $d(l_{w_i}, l_t)$ is the Euclidean distance and D is the diameter of a given area.

$f(w_i, t)$ is a function to calculate the discount to the worker's reputation and location proximity as a result of his/her available time $\Delta t(t, w_i)$. It is defined as:

$$f(w_i, t) = \frac{\Delta t(t, w_i)}{\max_{w_i \in W} \{\Delta t(t, w_i)\}} \text{ where } W \text{ is a worker set.} \quad (6)$$

Thereafter, based on the $CS(t, w_i)$ of worker $w_i (1 \leq i \leq m)$ from W with size m , the $ACS(t, W)$ could be calculated as follows:

$$ACS(t, W) = \sum_{k=\min_{W_i} |A \cap F_k|}^{|W|} \sum_{w \in A} \prod_{w \in A} CS(t, w) \prod_{w \notin A} (1 - CS(t, w)). \quad (7)$$

where F_k is all the subsets of W with size k

4 Assignment Protocol

In this section, we propose two two-phase greedy algorithms for QBTA problem. The first algorithm (Quality_Sensitive) is proposed to find out a worker set with higher quality, while another algorithm (Cost_Sensitive) is developed to find out a worker set with less total rewards.

4.1 The Quality_Sensitive Algorithm

The two-phase greedy algorithm namely Quality_Sensitive is proposed for QBTA problem, which is shown in Algorithm 1.

In the first phase (lines 1–4), it traverses the worker set W to select the workers who satisfy both $\Delta t(t, w) \geq 0$ and $rk(w) = H/M$, and then inserts them to $W'(t)$ in the descending order of $CS(t, w)$. In the second phase (lines 5–20), it first selects the top- K workers of $W'(t)$ to constitute the original W^* and the other workers constitute W_{re} . Then it needs to decide whether W^* satisfies the $CR_t(conf_t, minW_t)$ and $B(t)$ or not. There may incur three different conditions to be solved: (i) W^* does not satisfy the $CR_t(conf_t, minW_t)$. It will start the next loop. (ii) W^* satisfies $CR_t(conf_t, minW_t)$ and $B(t)$, it will return W^* . (iii) W^* only satisfies $CR_t(conf_t, minW_t)$. It will first find out the high-rank worker w_a with lowest CS in W^* and the medium-rank worker w_b with highest CS in W_{re} , then exchange them to generate the new W^* and W_{re} , finally, it returns to execute the eighth step in the current loop.

Algorithm1 Quality_Sensitive (t, W)

Input: a spatial task t , a worker set W

Output: $W^*(t)$

```

1: for each worker  $w \in W$  do
2:   if  $w$  satisfies that  $\Delta t(t, w) \geq 0$  and  $rk(w) = H$  or  $M$ 
3:     compute  $CS(t, w)$  based on the Eq. (4)
4:     insert  $w$  into  $W'(t)$  in the descending order of the  $CS(t, w)$ 
5:  $W^* = \Phi$ ;  $W_{re} = \Phi$ 
6: for  $K = minW_t$  to  $|W'(t)|$ 
7:    $W^* = \{w_1, w_2, \dots, w_K\}$  and  $W_{re} = \{w_{K+1}, w_{K+2}, \dots, w_{|W'(t)|}\}$ 
8:   if  $W^*$  does not satisfy the  $CR_t(conf_t, minW_t)$ 
9:     continue; /*stop the current loop and start the next loop*/
10:  else if  $W^*$  satisfies the  $B(t)$ 
11:    return  $W^*$ ;
12:  else
13:     $w_a \leftarrow$  the high-rank worker with lowest  $CS$  in  $W^*$ 
14:     $w_b \leftarrow$  the medium-rank worker with highest  $CS$  in  $W_{re}$ 
15:    if  $w_a$  and  $w_b$  are both exist
16:       $W^* \leftarrow W^* \cup \{w_b\} - \{w_a\}$ 
17:       $W_{re} \leftarrow W_{re} \cup \{w_a\} - \{w_b\}$ 
18:      continue to execute start from the eighth step
19:    else
20:       $W^* = \Phi$ 
21: end

```

Let us consider the example of Fig. 1. It first realigns the workers in the descending order of $CS(t, w)$ and then choose the top-3 workers w_3, w_4, w_6 to constitute W^* . However, $\{w_3, w_4, w_6\}$ does not satisfy the $CR(0.6, 3)$, thus $K = 4$, $\{w_3, w_4, w_6, w_8\}$ satisfies the $CR(0.6, 3)$ and $B(t)$. Therefore, $W^* = \{w_3, w_4, w_6, w_8\}$.

For the task t , the algorithm needs to traverse the worker set W for one time to generate the ordered $W'(t)$. Thereafter, for $\min W K \leq |W'(t)|$, it greedily chooses the top- K workers to constitute the W^* and then decides W^* whether satisfies $CR_t(conf_p, minW_t)$ and $B(t)$ or not. Therefore, the time complexity is $O(|W|)$.

4.2 The Cost_Sensitive Algorithm

Considering the task publisher always prefers to spend less cost for workers' rewards. Based on the Quality_Sensitive algorithm, another two-phase greedy algorithm namely Cost_Sensitive is proposed to return a worker set with less total rewards.

Given a spatial task t and a worker set W , it first inserts the workers who satisfy both $\Delta t(t, w) \leq 0$ and $r(w) = H/M$ to $W'(t)$ in the ascending order of the reward. Then it gives higher priority to the workers with higher CS among those with equal rewards. In the second phase, it will decide whether W^* satisfies both $CR_t(conf_p, minW_t)$ and B_t or not. There may incur three different conditions to be solved: (i) W^* does not satisfy $B(t)$, the algorithm must stop. (ii) W^* satisfies $CR_t(conf_r, minW_t)$ and $B(t)$, so it will return W^* . (iii) W^* only satisfies $B(t)$, it will first find out the worker w_a with lowest CS in W^* and the worker w_b with highest CS in W_{re} whose $CS(t, w_b) > CS(t, w_a)$, then exchange them to generate the new W^* and W_{re} , finally, it will continue to execute the thirteenth step in the current loop. The time complexity is $O(|W|)$.

5 Performance Evaluation

In this section, we first discuss the experimental setting and then present the experimental results.

5.1 Experimental Setting

The real check-in dataset are obtained from Gowalla [16], a location-based social network, where users check in at different hotels in their vicinities. We collect the check-in records over one week from Oct/9/2010 to Oct/15/2010 with 38,940 records, 2,226 users and 30,242 hotels. Then we partition it into 56 sub-datasets with a three-hour time range, based on the necessary of constructing the experimental data. Each check-in record indicates the location and time that user have entered the hotel.

For each sub-dataset, spatial taskset is generated based on the check-in records with the same hotel. The end of the time range is used as the expiration time of the task. The other attributes including $minW$, B and $conf$ are randomly selected in the range [4, 7], [25, 75], [0.5, 0.7], respectively. The corresponding worker set is constituted by the earliest check-in record of each user, and the location is used as the worker's current location. We measure the travel time as the Euclidean distance divided by the average travel speed (i.e., 30 miles/h). Additionally, each worker has a reputation parameter which is randomly selected in the range [0, 1].

To save the space, we only list the number of spatial tasks (T), workers (W), high-rank workers (HW), medium-rank workers (MW) in total for six sub-datasets in Table 1 and show their experimental results in the next subsection.

Table 1. The details for six sub-datasets

Sub_dataset	t1	t2	t3	t4	t5	t6
T	653	2498	2116	981	1242	2338
W	766	533	451	450	432	376
HW	102	61	66	170	179	149
MW	200	121	101	150	124	113

5.2 Experimental Results

Budget-TASC [3] is an algorithm which figures out a worker set with the maximum expected quality of the result and the total rewards within a pre-specified budget. We compare it with our two algorithms in terms of the runtime, the total ACS (*total_ACS*) and the total costs on workers' rewards (*total_rew*).

Runtime: This metric is used to measure how efficient of an algorithm. Figure 2 (left) shows the runtime of the three algorithms on six sub-datasets. *Cost_Sensitive* performs better than *Quality_Sensitive* because *Quality_Sensitive* always calculates the ACS first, which is time-consuming. *Budget_TASC* performs best on three sub-datasets (i.e., t4, t5 and t6) because there are more high-rank workers than medium-rank workers, which will effectively reduce the number of worker combinations and thus decrease the runtime.

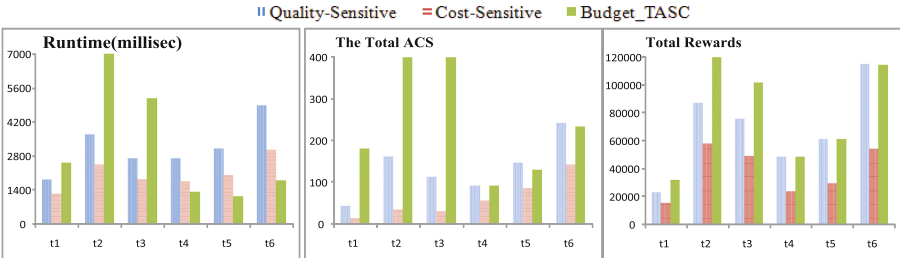


Fig. 2. The runtime, total ACS and total rewards for six sub-datasets

total_ACS: This metric is used to measure how effective of an algorithm, which is calculated in Eq. (8). Figure 2(middle) shows the *total_ACS* for three algorithms. *Quality_Sensitive* performs better than *Cost_Sensitive*, because it will find out the worker set for each task with least size and a higher ACS than others with same size for QBTA problem. But it may do not find the worker set with highest ACS. Thus, *Budget_TASC* performs better than *Quality_Sensitive* on t1, t2 and t3.

$$total_ACS(T) = \sum_{t \in T} ACS(t, W * (t)) \text{ where } T \text{ is a spatial taskset.} \quad (8)$$

total_rew: This metric is used to measure how effective of an algorithm, which is calculated in Eq. (9). Figure 2(right) shows the *total_rew* of three algorithms on six sub-datasets. Cost_Sensitive performs better because it will find out the worker set for QBTA problem with least size and a lower *total_rew* than others with same size.

$$total_rew(T) = \sum_{t \in T} \sum_{w \in W*(t)} rew(w) \text{ where } T \text{ is a spatial taskset.} \quad (9)$$

6 Conclusions

With the rapid development of communication technologies, spatial crowdsourcing has become more and more popular. As workers may possess different capabilities, a major challenge is how to assure the quality of results while considering the limit budget. This paper focuses on the issue of quality assurance for task assignment while considering budget limitation and then proposes two two-phase greedy algorithms. Finally, experimental results show that Quality_Sensitive gets the worker set with higher quality and Cost_Sensitive gets a worker set with less cost. Also, their less runtime makes them interactive and suitable for running on mobile platform.

Acknowledgments. This work is supported by National Natural Science Foundation of China under Grant No. 61572295; Innovation Method Fund of China No. 2015IM010200; Natural Science Foundation of Shandong Province under Grant No. ZR2014FM031; Science and Technology Development Plan Project of Shandong Province No. 2014GGX101047, No. 2015GGX101007, No. 2015GGX101015; Shandong Province Independent Innovation Major Special Project No. 2015ZDJQ01002, No. 2015ZDXX0201B03.

References

1. Kazemi, L., Shahabi, C., Chen, L.: GeoTruCrowd: trustworthy query answering with spatial crowdsourcing. In: ACM SIGSPATIAL, Orlando, FL, USA, pp. 304–313 (2013)
2. Deng, D., Shahabi, C., Demiryurek, U.: Maximizing the number of worker’s self-selected tasks in spatial crowdsourcing. In: ACM SIGSPATIAL, Orlando, FL, USA, pp. 314–323 (2013)
3. Yu, H., Miao, C., Shen, Z., Leung, C.: Quality and budget aware task allocation for spatial crowdsourcing. In: AAMAS, pp. 1689–1690 (2015)
4. Kazemi, L., Shahabi, C.: GeoCrowd: enabling query answering with spatial crowdsourcing. In: ACM SIGSPATIAL, Redondo Beach, CA, USA, pp. 189–198 (2012)
5. Gao, D., Tong, Y., She, J., Song, T., Chen, L., Xu, K.: Top-k team recommendation in spatial crowdsourcing. In: WAIM, pp. 191–204 (2016)
6. Xie, X., Chen, H., Wu, H.: Bargain-based stimulation mechanism for selfish mobile nodes in participatory sensing network. In: SMAHCN, pp. 72–80 (2009)
7. Li, Y., Yiu, M.L., Xu, W.: Orient online route recommendation for spatial crowdsourcing task workers. In: SSTD, pp. 137–156 (2015)

8. Kittur, A., Nickerson, J.V., Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J., Lease, M., Horton, J.: The future of crowd work. In: CSCW, pp. 1301–1318 (2013)
9. Sakurai, Y., Okimoto, T., Oka, M., Shinoda, M., Yokoo, M.: Ability grouping of crowd workers via reward discrimination. In: HCOMP, pp. 147–155 (2013)
10. Liu, Y., Zhang, J., Yu, H., Miao, C.: Reputation-aware continuous double auction. In: AAAI, pp. 3126–3127 (2014)
11. <http://www.uber.com>
12. Khosravifar, B., Bentahar, J., Gomrokchi, M., Alam, R.: CRM: an efficient trust and reputation model for agent computing. In: KBS, pp. 1–16 (2012)
13. Yu, H., Shen, Z., Miao, C., An, B., Leung, C.: Filtering trust opinions through reinforcement learning. In: Decision Support Systems, pp. 102–113 (2014)
14. Fang, H., Guo, G., Zhang, J.: Multi-faceted trust and distrust prediction for recommender systems. In: Decision Support Systems, pp. 37–47 (2015)
15. Wahab, O.A., Bentahar, J., Otok, H., Mourad, A.: A survey on trust and reputation models for web services: single, composite, and communities. In: Decision Support Systems, pp. 121–134 (2015)
16. <http://snap.stanford.edu/data/loc-gowalla.html>