https://dx.doi.org/10.1016/j.dss.2018.03.010

Toward a real-time and budget-aware task package allocation in spatial crowdsourcing

Pengkun Wu [a, b], Eric W. T. Ngai [b, *], Yuanyuan Wu [a]

[a] School of Management, Harbin Institute of Technology, Harbin, China

[b] Department of Management and Marketing, The Hong Kong Polytechnic University, Hong Kong, China

Abstract: With the development of mobile technology, spatial crowdsourcing has become a popular approach in collecting data or road information. However, as the number of spatial crowdsourcing tasks becomes increasingly large, the accurate and rapid allocation of tasks to suitable workers has become a major challenge in managing spatial outsourcing. Existing studies have explored the task allocation algorithms with the aim of guaranteeing quality information from workers. However, studies focusing on the task allocation rate when allocating tasks are still lacking despite the increasing unallocated rates of spatial crowdsourcing tasks in the real world. Although the task package is a commonly known scheme used to allocate tasks, it has not been applied to allocate spatial crowdsourcing tasks. To fill these gaps in the literature, we propose a real-time, budget-aware task package allocation for spatial crowdsourcing (RB-TPSC) with the dual objectives of improving the task allocation rate and maximizing the expected quality of results from workers under limited budgets. The proposed RB-TPSC enables spatial crowdsourcing task requester to automatically make key task allocation decisions on the following: (1) to whom should the task be allocated, (2) how much should the reward be for the task, and (3) whether and how the task is packaged with other tasks.

Keywords: Spatial crowdsourcing; Task allocation algorithm; Task package; Incentive mechanism; Greedy algorithm; Reputation

* Corresponding author.

E-mail addresses: pengkunwu@126.com (P. Wu), eric.ngai@polyu.edu.hk (E. W. T. Ngai), wuyuanyuan@stu.hit.edu.cn (Y. Wu).

1. Introduction

Crowdsourcing, an Internet-scale community to outsource tasks [1], has generated considerable interest in the popular press [2, 3]. It has successfully helped requesters solve a wide range of video, stock photography, and even corporate research and development (R&D) tasks [4, 5]. Firms are likely to use this cost-effective business model to generate new ideas [6] and identify solutions to business problems [2, 7], and individual users have sufficient motivation to participate in crowdsourcing tasks [4].

Currently, crowdsourcing is classified in two forms, namely, online crowdsourcing based on web platforms [8, 9] and mobile crowdsourcing based on mobile apps [10, 11]. The main difference between online and mobile crowdsourcing is the working platform. With the development of location-based technology and the popularity of smartphones, sufficient location information can now be easily obtained [12], resulting in new and practical spatial crowdsourcing based on sensor devices [13-15]. Spatial crowdsourcing depends on web platforms or mobile apps and significantly considers the location of tasks and users [16-18]. It requires workers to be at the specific locations so as to complete the designated tasks. Spatial crowdsourcing is also referred to as location-aware crowdsourcing. The majority of previous research on crowdsourcing cannot be applied to the field of spatial crowdsourcing because of their unique characteristics, especially the huge requirements assigned on the locations of tasks and workers [19]. Crowdsourcing is classified into two forms in terms of whether the location of tasks and users is considered, specifically, traditional crowdsourcing that does not consider location information and spatial crowdsourcing that considers location information.

For most of traditional crowdsourcing tasks, such as logo design, requesters aim to search for suitable workers, and workers aim to find satisfied requesters. These crowdsourcing tasks are usually complex, and platforms assign sufficient time for requesters to find suitable workers and for workers to select the tasks they prefer. In this case, platforms serve as a link between requesters and tasks.

The complexities of most of spatial crowdsourcing tasks are significantly lower than those of traditional crowdsourcing tasks. However, the number of spatial crowdsourcing tasks has rapidly increased. Platforms cannot allocate time for workers to select the tasks they prefer because of the large number of tasks. Similar to real platforms like Uber [20] and another known state-of-the-art budget-aware task allocation approach for spatial crowdsourcing (Budget-TASC) [19], we do not enable the users to select the tasks they prefer, and the allocations of these spatial crowdsourcing tasks are determined by the platforms rather than the requesters or users. In this case, platforms take on the responsibility of task assignment, and need to rapidly allocate considerable spatial crowdsourcing tasks to suitable workers.

To complete the spatial crowdsourcing tasks, workers are required to travel to specific locations and incur extra transportation costs. Previous studies have concluded that the distance between a worker and the task location negatively affect the quality of the result [21, 22]. Recently, some task allocation algorithms that consider the locations of tasks and workers have been proposed to reduce the distance between tasks and workers and to control the budget [19, 22]. Most of the existing task allocation algorithms for spatial crowdsourcing tasks have focused on the expected result quality and limited budgets and tend to ignore the task allocation rates of spatial crowdsourcing tasks. In the real world, however, the unallocated rates of spatial crowdsourcing are very high. From the data provided by one Chinese company, the unallocated rate of the spatial crowdsourcing tasks is high at 37.485% [23]. The result quality and the task allocation rate are two important indicators of success when designing task allocation algorithms.

Improving the task allocation rate of spatial crowdsourcing tasks has become a difficult problem for platforms when designing task allocation algorithms [24]. Packaging the tasks is a good way to improve the task allocation rate. In the field of spatial crowdsourcing tasks, few task package algorithms have been proposed despite the low task allocation rate they produce.

In the present study, we focus on spatial crowdsourcing wherein workers are required to complete tasks in a given location. We focus on the task package allocation algorithm, in which a revised greedy algorithm incorporated into the spatial crowdsourcing system automatically allocates the task to the appropriate worker and calculates the corresponding reward for the task. This study aims to design an efficient task package allocation algorithm that not only maximizes the expected quality of collective results from selected workers, but also improves the task allocation rate of spatial crowdsourcing tasks.

To achieve our aim, we propose a real-time, budget-aware task package allocation for spatial crowdsourcing (RB-TPSC), which maximizes the expected quality of information from the workers under a limited budget. The proposed RB-TPSC also improves the task allocation rate by jointly considering the track records and distances of workers from the tasks. It helps spatial crowdsourcing requesters decide on the following issues: (1) to whom should the task be allocated, (2) how much should the reward be for the task, and (3) whether and how the task is packaged with other tasks.

In this paper, we advance the key contributions of our research in the following ways:

(i) To the best of our knowledge, our work is the first to focus on the task allocation rate of spatial crowdsourcing and attempt to improve the task allocation rate by designing a task allocation algorithm.

(ii) This is also among the first to propose a task package allocation for spatial crowdsourcing tasks. Although the task package has been mentioned previously, it has remained unexplored for new type of tasks, such as spatial crowdsourcing tasks.

(iii) We propose the RB-TPSC by jointly considering the distances from the tasks and the track records of workers to maximize the expected quality of information from the latter under a limited budget, and to improve the task allocation rate of spatial crowdsourcing tasks.

(iv) The RB-TPSC supports spatial crowdsourcing platforms wherein rewards for the tasks can be automatically determined by combining the characteristics of tasks and their distances from the selected workers, once the tasks have been allocated to these workers.

2. Literature review

2.1 Crowdsourcing

As the times develop and markets change, market form must also change to accommodate them [25]. In this section, we provide a brief review of the development of crowdsourcing, which we summarize in Fig. 1.

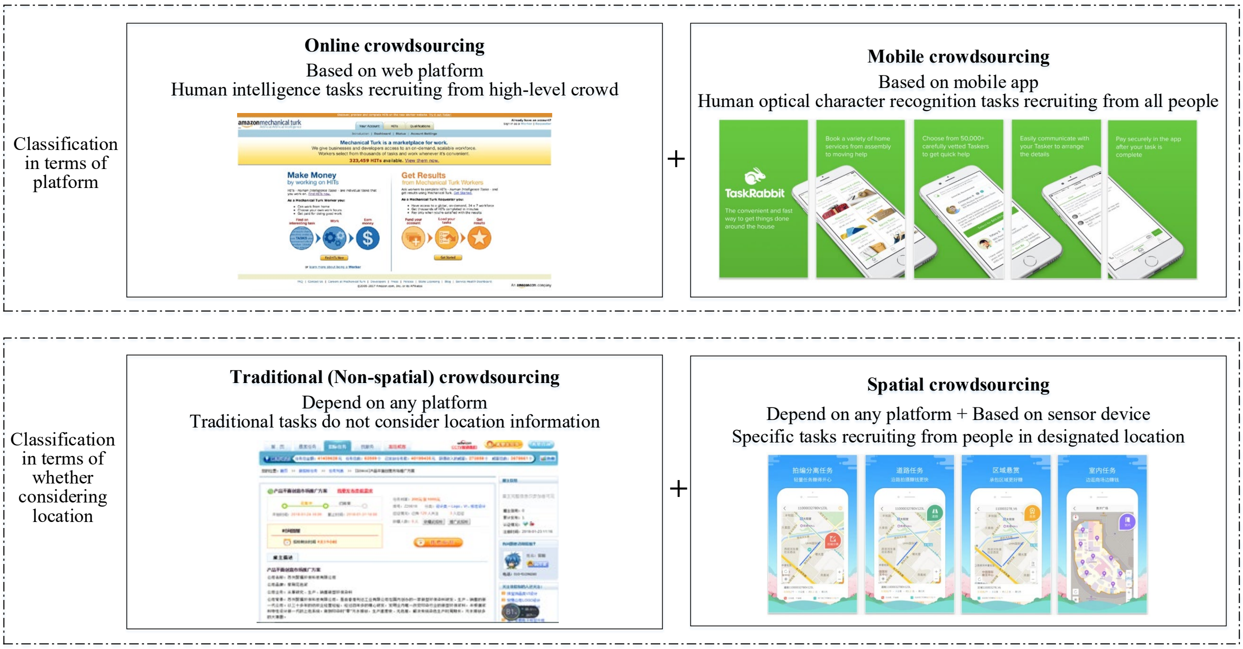
Classification in terms of platform
Online crowdsourcing
Based on web platform
Human intelligence tasks recruiting from high-level crowd
+
Mobile crowdsourcing
Based on mobile app
Human optical character recognition tasks recruiting from all people
TaskRabbit
Classification in terms of whether considering location
Traditional (Non-spatial) crowdsourcing
Depend on any platform
Traditional tasks do not consider location information
+
Spatial crowdsourcing
Depend on any platform + Based on sensor device
Specific tasks recruiting from people in designated location

Fig. 1 Classifications of crowdsourcing

The early examples of crowdsourcing tasks are based on web platforms, such as Amazon Mechanic Turk, Microworkers, Kickstarter, Topcoder, and TaskCN [26]. Based on these websites, requesters outsource some difficult tasks to collect required data, find suitable solutions, and seek excellent opinions from a large crowd of public participants [9]. Given that crowdsourcing can lower the R&D costs for firms seeking solutions without compromising quality [27, 28], firms have increasingly relied on crowdsourcing to outsource human intelligence tasks, such as creative design tasks and open innovation contests [1, 4, 28-34].

Recently, mobile apps have also adopted crowdsourcing to harness the power of the crowd to share relevant information or improve services [10, 11]. Mobile crowdsourcing is based on mobile apps, such as

TaskRabbit, Placemeter, Weather Signal, Wave, and Blablacar [35]. The high penetration of mobile devices demonstrates their impact on our daily lives [36]. By combining online and mobile crowdsourcing, requesters can outsource not only human intelligence tasks to high-level crowd, but also human optical character recognition tasks to ordinary people.

With the remarkable proliferation of built-in global positioning system (GPS) on smartphones [37] and the rising number of active mobile users [35, 38], requesters can now recruit workers speaking a particular language or living in a given city by outsourcing spatial crowdsourcing tasks, such as taking photos in a certain place [16-18]. Ubiquitous technologies, such as smartphones and public displays, are now mature enough to allow users to contribute to a wide range of crowdsourcing tasks regardless of time and location [14]. Various mobile devices and wearable technologies equip people with unprecedented computing and sensing capabilities, giving birth to the prevalence of spatial crowdsourcing [13-15].

Spatial crowdsourcing does not require excessive efforts from workers, thus reducing perceived barriers to participation [39]. Furthermore, spatial crowdsourcing allows for a geo-fenced and contextually controlled crowdsourcing environment; thus, certain individuals with certain location advantages can be targeted [40].

2.2 Incentive mechanism

Motivating human participation and improving the quality of contribution are crucial to the success of spatial crowdsourcing tasks [24]. Without sufficient incentives, workers are reluctant to receive any crowdsourcing tasks [41, 42]. On the one hand, the tasks with low reward cannot attract workers, resulting in the failure of task allocation. On the other hand, if the reward is high, firms cannot optimize their profits [43]. Crowdsourcing requesters must, therefore, design a suitable incentive mechanism closely related with task allocations [44, 45].

Incentive mechanisms can be broadly divided into two major types: monetary incentive mechanisms [46, 47] and gamification incentive mechanisms [48]. Despite the existence of crowdsourcing platforms that provide workers gamification incentives, using non-monetary incentives for tedious and repetitive crowdsourcing works remains difficult [24]. The main part of incentives must be monetary incentive, which significantly increases the interest of participants in reinforcing task quality [47].

Some researchers have conducted pioneering works on motivating workers to contribute their resources by designing monetary incentive mechanisms [49]. Many auction schemes have been proposed for traditional (e.g., non-spatial) crowdsourcing through the consideration of the behaviors and thoughts of workers, but the task allocations are similar and the quality of allocated tasks cannot be guaranteed [41]. The auction mode also disregards the impact of the spatial distribution of the locations between workers and tasks, and cannot be used in the field of spatial crowdsourcing [19]. Furthermore, spatial crowdsourcing tasks have high requirements concerning the ever-changing locations of tasks and workers. The number of workers located in a certain location may increase or decrease sharply depending may change at different time points. Thus, the incentive mechanisms for spatial crowdsourcing tasks should be in real-time [24, 46].

2.3 Task allocation algorithm

Many task allocation algorithms have been proposed for traditional (non-spatial) crowdsourcing tasks [19]. Based on network queueing theories, the authors usually considered the characteristics of workers to allocate tasks, especially their reputation values [50]. Some other studies have focused on the constraint of a limited budget. Bayesian learning [51] and Partially Observable Markov Decision Processes (POMDP)-based techniques [52] have been used to estimate the ground truth, but these studies disregarded the fact that different tasks should be given different rewards. Recursive reverse auction-based mechanisms are also used to mark the workers' bid for different rewards earned for performing different tasks [53]. The representative studies about task allocation for crowdsourcing tasks include the agent-based budget allocation algorithm (CrowdBudget) [54] and the Dual Task Assigner (DTA) [55], which allocate a specified budget through performance-cost analyses. However, these studies disregarded the impact of the spatial distribution of tasks and workers; thus, they cannot be applied to the allocation of spatial crowdsourcing tasks.

In comparison with traditional crowdsourcing, incentive mechanism design and task allocation algorithms for spatial crowdsourcing have received less research attention [19]. Some studies have found that workers mainly prefer solving tasks in close proximity of their locations [56], but such studies disregarded the characteristics of workers [21]. As representative works, the efficient heuristic-based greedy approach [22, 57], the fog-based deduplicated spatial crowdsourcing (Fo-DSC) framework[58], and the two-phase-based global online allocation (TGOA) [59] have been proposed to guarantee result quality by allocating the tasks to workers around tasks. However, these approaches failed to consider budget limitation.

With the aim of controlling the budget for all tasks, Budget-TASC [19], the budget limited crowdsourcing for interdependent task allocation with quality guarantees (BudgetFix) [60], and the adaptive budget algorithm (ADAPT) [61] have been proposed to allocate tasks by considering the characteristics of both workers and tasks as well as the limited budgets. These representative task allocation algorithms for spatial crowdsourcing tasks are summarized in Table 1. The Budget-TASC, the newest among these algorithms, outperforms other approaches as demonstrated by past experiments [19]. Thus, we compare our proposed RB-TRSC with the Budget-TASC.

Table. 1 Summary of representative task allocation algorithms for spatial crowdsourcing tasks

Studies	Algorithms	Considers result quality	Considers task allocation rate	Considers limited budgets
Kazemi et al. 2013	GeoTruCrowd	√	√	×
Feng et al. 2014	TRAC	×	×	√
Karger et al. 2014	Budget-optimal	√	×	√
Tran-Thanh et al. 2014	BudgetFix	√	×	√
Zhao et al. 2014	OMZ/OMG	√	×	√
Ni et al. 2016	Fo-DSC	√	×	×
Tong et al. 2016	TGOA	√	×	×
To et al. 2016	ADAPT	√	×	√
Miao et al. 2016	Budget-TASC	√	×	√

Most of these existing task allocation algorithms have focused on the result quality and limited budgets, but ignored the task allocation rate of spatial crowdsourcing tasks. In the real world, however, the unallocated rate of spatial crowdsourcing is very high. To fill these gaps in the literature, we proposed the RB-TPSC, which improves the task allocation rate of spatial crowdsourcing, and moreover it maximizes the expected quality of information from selected workers under a limited budget.

3. RB-TPSC approach

3.1 Definitions of symbols

For the convenience of the readers, the symbols used in this paper are listed in Table 2 in the order of their appearance.

Table. 2 Definitions of all notations used in the paper

Notation	Definition
	Sets
$n_i \in N$	Worker set
$m_j \in M$	Task set
$h_j \in H$	Undistributed task set that cannot be allocated by first task allocation
Q_i	Set for the tasks that can be completed by worker n_i
S	Set for the matching relationship between tasks and corresponding selected workers
	Subscripts
i	Subscript to describe the spatial crowdsourcing worker n_i
j	Subscript to describe the spatial crowdsourcing task m_j
j_i	Subscript to describe the specific spatial crowdsourcing worker selected for task m_j
	Symbols
A_j	Priority of task m_j
C_j	Extra monetary incentive that the requester of task m_j pay to add the priority
R_j	Radiation radius of task m_j
α	Reward for unit time

t_j	Elapsed time of task m_j
F_j	Baseline reward for task m_j
ER	Earth radius
(x_i, y_i)	Location (longitude and latitude) of worker n_i
(x_j, y_j)	Location (longitude and latitude) of task m_j
d_{ij}	Distance between worker n_i and task m_j
B_j	Budget for task m_j
β	Extra remote subsidy per kilometer
γ	Accepted distance without extra remote subsidy
$d_{j_i j}$	Distance between task m_j and the selected worker n_{j_i}
E_j	Extra remote subsidy for task m_j
P_j	Reward for task m_j
v_i	Reputation value of worker n_i
q_i	Predetermined quota of worker n_i
k	Index for number of allocated tasks in the algorithms
u_i	Number of accepted tasks of worker n_i
nb	Number of tasks completed under both the tested approach and real-life approach
nr	Number of tasks completed in real-world
	Measurement indicators
ρ	Matching-degree with the real-world data
η	Completion rate of spatial crowdsourcing tasks
ψ	Average reputation value of all workers selected to complete tasks
ζ	Average distance travelled
ϕ	Average budget utilization rate
ω	Average reward

3.2 Problem formulation

This study focuses on the problem of efficient task package allocation, which emerged from many real-world applications. One of the significant is the car-sharing. Ordinary people become the requesters as they outsource their travel plans, whereas the car owners become the workers. The mobile platform collects various data and depend on these data to allocate the tasks to the suitable car owners. Other representative examples include taking photos and collecting information on a certain place. Traditionally, when requesters need to obtain some information or photos of a certain place, they must pay a high cost to allocate employees to a specific place and perform the required tasks. With the remarkable proliferation of intelligent wireless devices and active mobile users, requesters can now outsource these spatial crowdsourcing tasks, and users on a certain place can depend on the reward and task information to decide whether they are interested in the tasks. Thus, we formally define our research problem on task allocation.

Problem formulation 1 (task allocation): *Given a set of users* $N=\{n_1,n_2,\mathrm{L}\ ,n_i,\mathrm{L}\ \}$ *and a set of spatial crowdsourcing tasks* $M=\{m_1,m_2,\mathrm{L}\ ,m_j,\mathrm{L}\ \}$. *For task* m_j, *its priority* A_j *is decided by the extra monetary incentive* C_j *that the requester is willing to pay. Let* $Q=\bigcup_{i=1}^{|N|}Q_i$ *be the union of potential match sets for all users, and every match* Q_i *is of the form* $\left(n_i,\left\langle m_{j1},m_{j2},m_{jk},\mathrm{L}\ \right\rangle\right)$, *in which* $\left\langle m_{j1},m_{j2},m_{jk},\mathrm{L}\ \right\rangle$ *includes all tasks* m_j, *whose distance from user* n_i *is lower than the radiation radius* R_j. *To maximize the expected quality of collective information, we define users as being ordered by their reputation values. Each user selects the optional task with the shortest distance from the user's current location.*

Based on the task allocation algorithm, we present the task allocation set S, where tasks are assigned to workers within the radiation radius, within which they can select the tasks close to them. Because of the low allocation rates of many spatial crowdsourcing tasks [23], improving the task allocation rate of spatial crowdsourcing tasks has become a new problem for platforms when designing algorithms to allocate tasks [24]. Packaging the tasks is a good way to improve the task allocation rate. We thus revise our research problem and formally define the research problem on task package allocation.

Problem formulation 2 (task package allocation): *Given a set of users* $N=\{n_1,n_2,\mathrm{L}\ ,n_i,\mathrm{L}\ \}$ *and a set of spatial crowdsourcing tasks* $M=\{m_1,m_2,\mathrm{L}\ ,m_j,\mathrm{L}\ \}$. *For task* m_j, *its priority* A_j *is decided by the extra monetary incentive* C_j *that the requester is willing to pay. Let* S *be the initial match sets for allocating tasks to suitable workers from the task allocation algorithm, and* H *be the set including the tasks* *belonging to set M but not belonging to set S. To maximize the task allocation rate of tasks, we define users as being ordered by their reputation values. Each user selects the optional task with the shortest distance from their current location and the optional tasks in set H.*

3.3 Incentive mechanism

Prior to allocating tasks to suitable workers, we first design the incentive mechanism, which is closely related to task allocation. The platform designs the reward for unit time α, which is consistent with the local average wage. When requesters plan to post the spatial crowdsourcing task m_j, they should have an initial estimation of the elapsed time t_j. The reward for unit time multiply by the elapsed time is the baseline reward for task m_j. The calculation formula for the baseline reward for task m_j is shown in Eq. (1)

$$F_j = \alpha t_j. \tag{1}$$

Through advanced technology, we can obtain accurate distances between the tasks and the users. Thus, actual distances are used when the proposed algorithm is used in the real world. Similar to several research papers, our actual data did not include accurate distances. Thus, we calculated the distances based on known longitudes and latitudes for all tasks and users in environments without accurate distances. When all tasks and users are located in the Northern and Eastern Hemisphere, the Euclidean distance d_{ij} between task m_j and user n_i is calculated through Eq. (2), where ER refers to the earth radius [62].

$$\begin{cases} d_{ij} = ER * Arc\cos(C) \\ C = \cos(y_i * \pi/180) * \cos(y_j * \pi/180) * \cos\left((x_i - x_j) * \pi/180\right) + \sin(y_i * \pi/180) * \sin(y_j * \pi/180) \end{cases}. \tag{2}$$

When posting the spatial crowdsourcing tasks, the requester specifies the requirements that the workers must meet as they perform the tasks. The requirements usually include the location of the expected worker and the accepted radiation radius described as R_j, which is jointly decided by the budget for task m_j and basic reward of task m_j. The platform refuses to search for a worker if the cost that the requester can provide is lower than the baseline cost. The higher the costs the requester is willing to pay, the larger the radiation radius will be. Here, R_j is computed by using Eq. (3)

$$R_j = \begin{cases} \dfrac{B_j - F_j}{\beta} + \gamma \ , \ B_j \geq F_j \\ 0 \qquad\qquad\ , B_j < F_j \end{cases} , \tag{3}$$

where the budget for task m_j, extra remote subsidy per kilometer, and accepted distance without extra remote subsidy are represented by B_j, β and γ, respectively. All workers within the radiation radius R_j are added into candidate set for task m_j. We acquire Q_i, the set for the tasks that can be allocated by worker n_i, by transforming the candidate set. Then, worker n_i chooses the closest task from all selectable tasks in set Q_i. If the task requesters select the workers without γ, they must pay extra transportation fee E_j, which is expressed in Eq. (4)

$$E_j = \begin{cases} 0 \qquad\qquad\ , d_{j_i j} \leq \gamma \\ \beta\left(d_{j_i j} - \gamma\right) , d_{j_i j} > \gamma \end{cases} . \tag{4}$$

Thus, once the tasks have been allocated to suitable workers, the corresponding rewards for all tasks are automatically calculated by adding the extra remote subsidy and the extra monetary incentive to the basic reward. The expression for the reward of task m_j is shown in Eq. (5)

$$P_j = F_j + E_j + C_j . \tag{5}$$

3.4 First stage of RB-TPSC: task allocation algorithm

Given the reputation and location information of workers, the objective is to design an effective task allocation plan on behalf of the platform with the aim of improving the task allocation rate and maximizing the expected quality of information from workers under limited budgets. We first design a task allocation to allocate tasks to suitable workers that maximizes the expected quality of information from workers under limited budgets and screen out the undistributed tasks. Then, by focusing on those remaining tasks, we design the task package algorithm to obtain the final task allocation scheme to improve the task allocation rate.

The problem of the first stage of task allocation is the optimization, the objective of which is to maximize the expected quality of information from workers under limited budgets. According to past research, in the case of spatial crowdsourcing tasks, the quality of the obtained result depends on two main factors: the intrinsic trustworthiness of workers as reflected by their reputation values [63], and the distances between the workers and the tasks at hand [19]. Workers with good reputation values generally provide more reliable results than those with low reputation values [50, 64], and the distance between the selected worker and the task negatively affects the likelihood that the task requester receives quality results from the worker [21, 22, 57]. In this study, two heuristics are used to reduce the complexity of the problem. First, when their distances from the task are the same, the workers with high reputation values provide results of high quality. Second, when they have the same reputation values, the workers close to the tasks provide results of high quality [19].

Thus, the objective of the first stage can be expressed in two objectives: to minimize the average distance between the tasks and the selected workers, and to maximize the average reputation values of the selected workers. The objective functions are expressed in Eq. (6)

$$\begin{cases} \min \dfrac{1}{|S|}\sum_{j\in S} d_{j_i j} \\ \max \dfrac{1}{|S|}\sum_{j\in S} r_{j_i} \end{cases}, \tag{6}$$

where $|S|$ represents the cardinality of the set S.

To guarantee the task allocation quality, we also limit the maximum number of tasks that one worker can accept as q_i, which is also called predetermined quota of worker n_i. Besides affected by the task characteristics, the predetermined quota q_i is positively related to a worker's reputation value v_i, which is decided by the platform.

The problem of spatial crowdsourcing task allocation is a complex Multiple Choice Knapsack Problem (MCKP). To solve the problem, we designed the RB-TPSC, the first stage of which is shown in Algorithm 1. The workers are ordered by their reputation values and the highly ranked ones have high priorities in the process of task selection.

Algorithm 1: First stage of RB-TPCS: **task allocation algorithm**

Input: Workers $n_i \in N$, Tasks $m_j \in M$.

Output: The set of matching relationship between workers and tasks - S.

```
$S \leftarrow \varnothing$;
$k = 0$; $u_i = 0$;
Obtain necessary parameters, including $R_j$;
Sorting all the tasks in descending order based on the their priorities $A_j$;
For task $m_j \in M$ according to the obtained order
    For worker $n_i \in N$
        If worker $n_i$ is within the radiation radius $R_j$ for task $m_j$
            Put task $m_j$ into set $Q_i$;
        End
    End
End
Sorting all the workers in descending order based on the reputation values $v_i$;
For worker $n_i \in N$ according to the obtained order
    If there exist selected tasks in set $Q_i$ but not in set $S$
        Select the task $j$ with shortest distance from worker $n_i$;
        Calculate the reward for task $j$: $P_j = F_j + E_j + C_j$;
        Set: $k = k + 1$; $u_i = u_i + 1$; $S(k,:) = (k, n_i, m_j, P_j, d_{ij})$;
    End
End
Return set $S$.
```

3.5 Second stage of RB-TPSC: task package allocation algorithm

In the first stage of the RB-TPSC, we designed an effective task allocation scheme to allocate tasks, with the aim of maximizing the expected quality of information from workers under limited budgets. However, aside from this goal, we should improve task allocation rate of spatial crowdsourcing tasks. The real-world data from a Chinese company show that the task allocation rate of its spatial crowdsourcing tasks is only 62.515% [23].

To improve the task allocation rate, we emphatically focused on those undistributed tasks in the second stage of the RB-TPSC. All undistributed tasks left in algorithm 1 are put into set H. For each undistributed task h_j, we find the closest worker and then evaluate whether the distance is lower than its radius R_j. If the shortest distance is lower than R_j, then this undistributed task can be allocated by the corresponding worker. We package this undistributed task and other tasks allocated by the same worker together. If the shortest distance is larger than R_j, we can give up this task. Through the second stage, we can maximize the task allocation rate of the spatial crowdsourcing tasks. The second stage of the RB-TSPC is shown in Algorithm 2.

Algorithm 2: Second stage of RB-TPCS: **task package allocation algorithm**

Input: Workers $n_i \in N$, Tasks $m_j \in M$, undistributed tasks $h_j \in H$, initial matching set S, and the output parameters obtained from Algorithm 1, including k.

Output: Updated set of matching relationship between workers and tasks - S.

```
1. Sorting all the undistributed tasks in descending order based on the their priorities $A_j$;
2. For undistributed task $h_j \in H$ according to the obtained order
3.     Index $l = 0$;
4.     For worker $n_i \in N$
5.         Set: $l = l + 1; dist(l) = d_{lj}$;
6.     End
7.     Find the minimum component of the vector $dist$;
```

```
        Let the minimum value and corresponding worker be $dm$ and $ci$, respectively;
        If $dm \le R_j$ and $u_i < q_i$
            Choose to be the worker $ci$ for task $h_j$;
            Package task $h_j$ and other tasks selected by worker $ci$;
            Set: $k = k+1; u_i = u_i + 1; S(k,:) = (k, ci, h_j, P_j, d_{ij})$;
      End
End
Return set $S$.
```

4. Experimental evaluation

4.1 Dataset

To test the proposed RB-TPSC under realistic settings, we used the newest dataset[1] of 835 sharing-tasks involving 1877 users [23]. These 835 tasks are mainly obtained from four cities in China: Shenzhen, Guangzhou, Dongguan, and Foshan. The distributions of these 835 tasks are described in Fig. 2, in which the allocated and unallocated tasks are marked in red ("√") and blue ("×") markers, respectively. From Fig. 2, we can see that the original task allocation rate of these spatial crowdsourcing tasks is low. In addition to the locations, the dataset also provides the reputation values and predetermined task quota of all workers.

[1] The public dataset can be found in: http://www.mcm.edu.cn/html_cn/node/460baf68ab0ed0e1e557a0c79b1c4648.html.

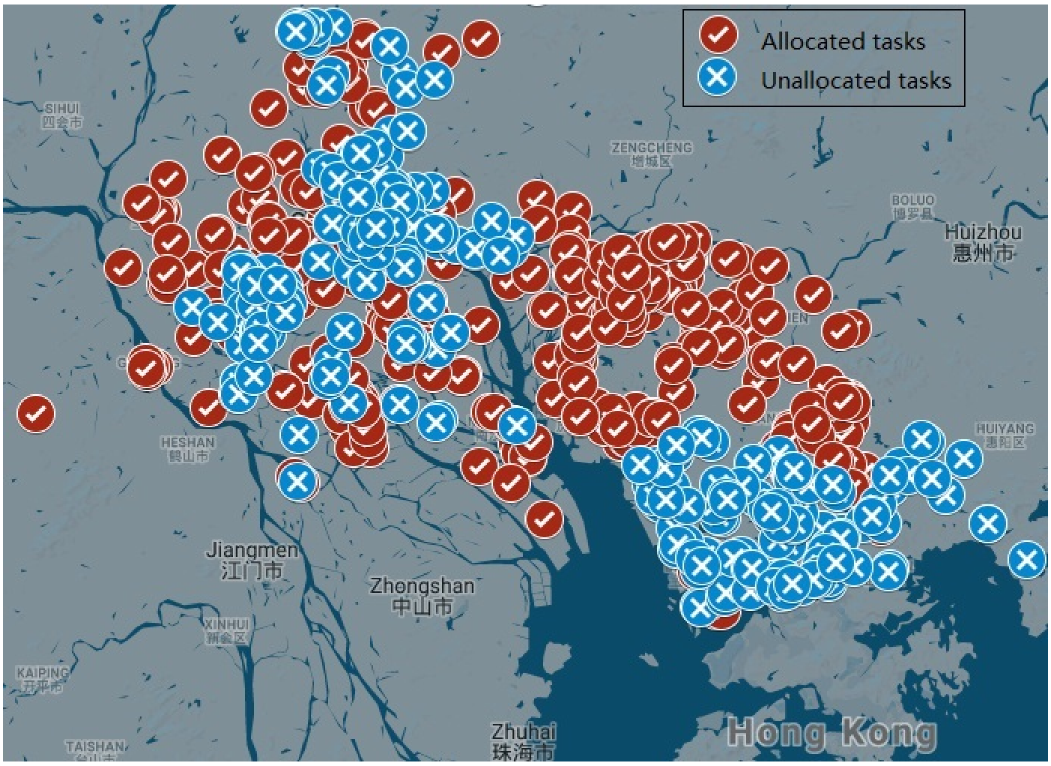

Fig. 2 Geo-spatial distribution of all tasks in the dataset

4.2 Experimental design

During this experiment, we first compare the proposed RB-TPSC with Budget-TASC [19]. For the Budget-TASC, the parameters that should be confirmed in advance include D_c, P_H, P_M, P_L, Th_{HM}, and Th_{ML}. In this experiment, we set the parameter $D_c = 68.4940\ km$, which is the average value of the diameters of the four task-issued cities. The diameters of Guanzou, Shenzhen, Dongguan, and Foshan are 97.2896, 50.4229, 56.0226, and 70.2410 km, respectively. We use Th_{ML} and Th_{HM} to refer to the average reputation value (278.1344) and the average reputation value (3871) of workers, whose reputation value is higher than Th_{ML}, respectively. Given that the original rewards for all tasks roughly present the uniform distribution from 65 to 85, we set $P_L = 65$, $P_M = 75$, and $P_H = 85$.

Next, we design a simulation by varying the parameter setting in order to understand the proposed RB-

TPSC and to improve its application in the real-world. The following three parameters are adjusted in the experiments to simulate different scenarios.

1. Extra remote subsidy per kilometer (β): In this experiment, the parameter varies from 0–20 monetary units in 1-unit increments (i.e., 21 different settings).
2. Accepted distance without extra remote subsidy (γ): In this experiment, the parameter varies from 0–2 km in 0.1-km increments (i.e., 21 different settings). The shortest distances between tasks and workers are shown in Fig. 3, from which we can see that most of the tasks can be reached by workers within 1 km and that the average shortest distance is 0.9109 km. Among the 835 tasks, just 2 tasks are considered far from all these workers, and only about 10 tasks are far from all workers with the shortest distance above 5 km.
3. Budget for task (B_j): For different tasks, the budget values are different and are predetermined by task requesters. In this experiment, the budgets change with a common percentage varying from 90%–110% in 1% increments (i.e., 21 different settings).

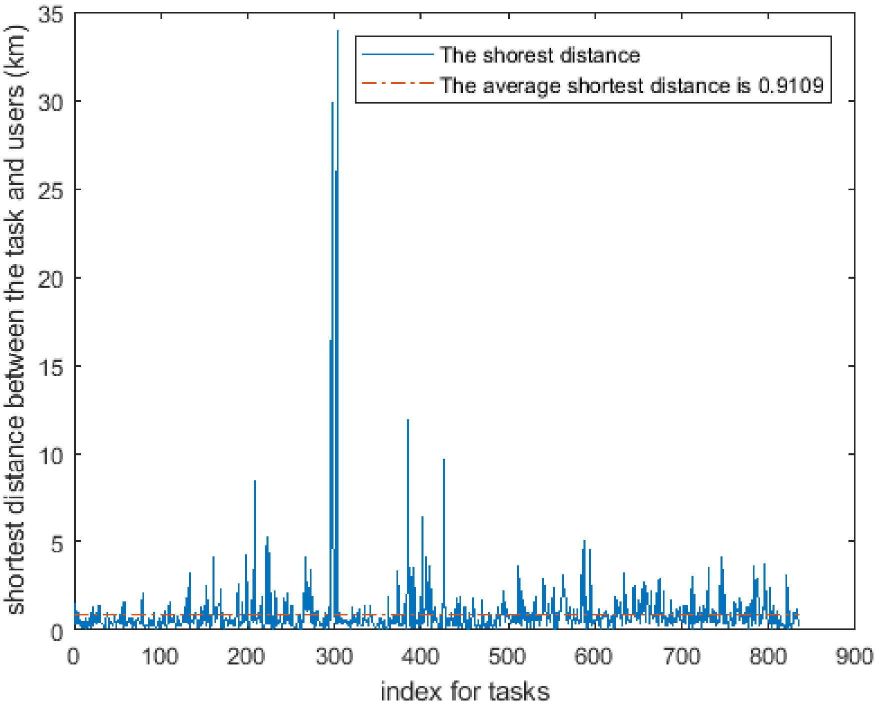

Fig. 3 Shortest distance between these tasks and all users

In the experiment, we measured the performance of each approach using the following six metrics.

1. Matching-degree with the real-world data (ρ): Before testing the proposed approach, we first analyze the matching degree between our results and the real-world data. Based on either our task allocation approach or the real-life approach, most of the tasks can be allocated. We test the matching degree using the statistics on the proportion of the number of tasks allocated under both the tested approach and the real-life approach (nb) compared with the number of tasks allocated in the real-world (nr).

$$\rho = \frac{nb}{nr} \tag{7}$$

2. Task allocation rate of spatial crowdsourcing tasks (η): This metric measures the effective approach that can encourage workers to complete the spatial crowdsourcing tasks. The task allocation rate η is the ratio of allocated tasks to the total number of spatial crowdsourcing tasks in an experiment.

$$\eta = \frac{|S|}{|M|} \tag{8}$$

3. Average reputation value of all workers selected to complete tasks (ψ): This metric is computed as the ratio of total reputation values of selected workers to the number of allocated tasks.

$$\psi = \frac{1}{|S|} \sum_{j \in S} r_{j_i} \tag{9}$$

4. Average distance travelled (ζ): This metric is computed as the average distance travelled by the selected workers for those allocated spatial crowdsourcing tasks.

$$\zeta = \frac{1}{|S|} \sum_{j \in S} d_{j_i j} \tag{10}$$

5. Average budget utilization rate (ϕ): This metric is computed as the average value of the ratio of actual reward for the allocated spatial crowdsourcing task to the budget of that task.

$$\phi = \frac{1}{|S|} \sum_{j \in S} \frac{P_j}{B_j} \tag{11}$$

6. Average reward (ω): This metric is computed as the average value of the actual reward for all allocated

spatial crowdsourcing tasks.

$$\omega = \frac{1}{|S|}\sum_{j\in S} P_j \tag{12}$$

The first indicator, matching-degree with the real-world data, is the foundation for the take next set of analyses. Related to the first objective, the second indicator measures the task allocation rates of all the tasks. The third and fourth indicators state the probability that our approach can obtain the expected results. The last two indicators are about the limited budget. From the perspective of task requesters, the first three indicators must be as high as possible, whereas the last three indicators must be as low as possible.

4.3 Comparison results

Among the existing task allocation algorithms in Table 1, the Budget-TASC outperforms other approaches [19]. Thus, we compared our proposed RB-TRSC with the Budget-TASC. Based on the real-world data, we calculate the six key metric values under these two approaches.

From the provided data set, we obtain the reputation values (v_i) and locations (x_i, y_i) of all workers and the budgets (B_j) and locations (x_j, y_j) of all tasks. Given that these 835 tasks almost have similar difficult coefficients and budgets ranging from 65 to 85, we set the basic rewards for these tasks as $F_j = 65$. Combined with the reality, we set the accepted distance without extra remote subsidy and the extra remote subsidy per kilometer as $\beta = 2$ and $\gamma = 0.5km$, respectively. These three parameters, β, γ, and B_j, are further analyzed through simulations in the next section.

As shown from the original data, among the 835 tasks, only 522 tasks have been allocated, indicating a task allocation rate of only 62.51%. The average cost for these 522 tasks is 69.8199. Based on our proposed RB-TPSC, the task allocation rate increases to 93.41%, whereas the average cost of allocated tasks decreases to 66.9003. The final task allocation schemes under the proposed RB-TPSC are consistent with the real world as allocated tasks can also be accomplished under our approach. The matching degree between our proposed

RB-TPSC and the real world is high 93.49%. Moreover, about 93.41% of spatial crowdsourcing tasks can be allocated by workers with the total average reputation value of 468.1863. The average distance travelled, average budget utilization rate, and the average reward for all tasks are 1.39 km, 0.9681, and 66.9003, respectively.

In comparison with the results under the first stage of RB-TPSC, the final results show that the task allocation rate significantly increases. When additional tasks can be allocated through packaged tasks, the average distance and average reward decrease despite the slight decrease in the average reputation value. These results indicate that packaging tasks can effectively improve the task allocation rate and guarantee the expected quality of information provided by workers.

We also compare the results under our proposed RB-TPSC and Budget-TASC. The main results under these two approaches are shown in Table 3. The proposed RB-TPSC clearly improves the task allocation rate better than the Budget-TASC with a progress rate of over 8.62%. The proposed RB-TPSC also has better expected quality of information from the workers because it outperforms the Budget-TASC by over 266.45% in terms of an increase in the average reputation value. Despite a slight increase in the average budget utilization rate and average reward compared with the Budget-TASC, the proposed RB-TPSC can help the selected workers travel less distances.

Table.3 Main results under different approach

Approaches	k (+)	ρ (+)	η (+)	ψ (+)	ζ (km) (-)	ϕ (-)	ω (-)
Original results in real-world	522	/	62.51%	/	/	/	69.8199
Budget-TASC	718	86.97%	85.99%	127.7625	1.521	0.9426	65.4039
First Stage of RB-TPSC	662	77.97%	79.28%	528.992	1.4411	0.9678	67.024
RB-TPSC	**780**	**93.49%**	**93.41%**	**468.1863**	**1.39**	**0.9681**	**66.9003**

4.4 Simulation results

By simulating and analyzing the results under different parameters, we can further understand the proposed

RB-TPSC and then apply it to the real-world. Three parameters, namely, extra remote subsidy per kilometer β , accepted distance without extra remote subsidy γ , and budget for task B_j are modified in the experiments to simulate different scenarios.

In the first experiment, the extra remote subsidy per kilometer (β) varies from 0–20 monetary units in 1-unit increments. We obtain the main results under different values of extra remote subsidy per kilometer, as shown in Fig. 4. As can be clearly seen, these six indicators have superior results when β=1. If $\beta > 1$, except the average distance travelled, the other five indicators are negatively affected by the value of extra remote subsidy per kilometer due to the inverse relationship between search radius and extra remote subsidy. Usually, the overall budget for the tasks is limited; hence, the task requester can only search for suitable workers within an extremely limited scope. If the extra remote subsidy is cancelled, we cannot obtain superior results because the task allocation rate is slightly lowered.

Thus, low extra remote subsidy cannot attract workers to complete tasks, whereas high extra remote subsidy limits the search scope of task requesters. The suitable value of extra remote subsidy should match the local economic level. Under this circumstance, the value of extra remote subsidy per kilometer must be β=1.

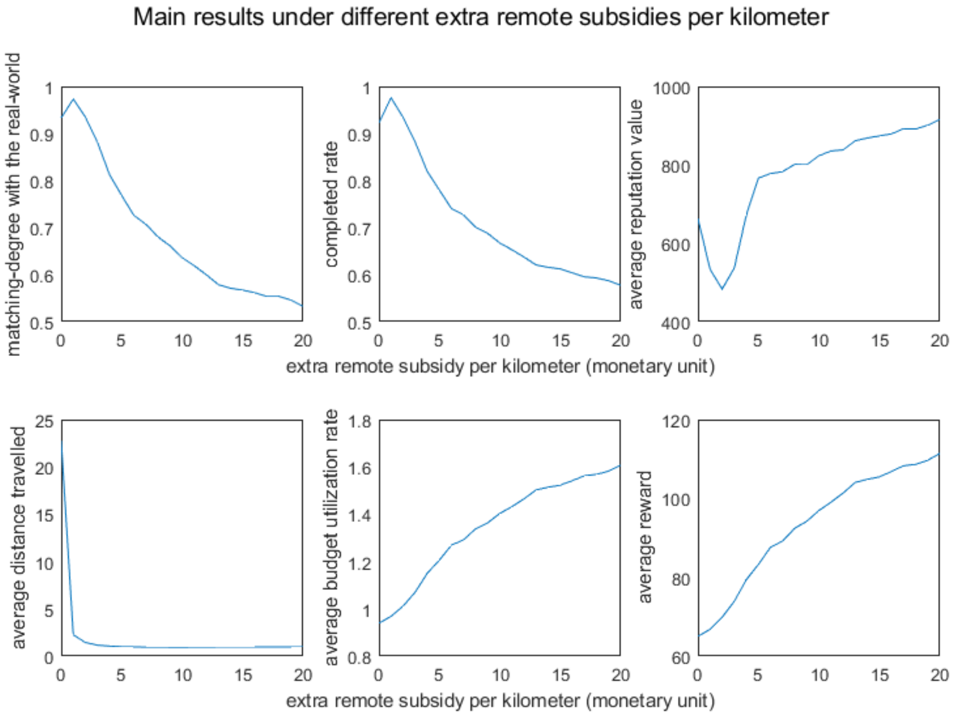

Fig. 4 Main results for different β values

In the second experiment, the accepted distance without extra remote subsidy (γ) varies from 0–2 km in 0.1-km increments. Under the designed circumstance, we obtain the main results under different values of accepted distances without the extra remote subsidy, as shown in Fig. 5. We find that these six indicators have excellent results when γ=2 . Except the average distance travelled, the results of five indicators show improvement when the value of accepted distances without extra remote subsidy increases. However, the changing range is extremely limited when $\gamma > 1$. The value of the average distance travelled decreases when $\gamma < 0.4$, and then increases sharply.

Although a large distance without the extra remote subsidy basically leads to improved results for task requesters, we note that the value should be mainly decided by the willingness of the workers. If workers have low willingness to complete tasks without the extra remote subsidy, then the value of γ must be 0.4. If workers have low willingness to complete tasks without the extra remote subsidy, then the value of γ can be larger than 1.

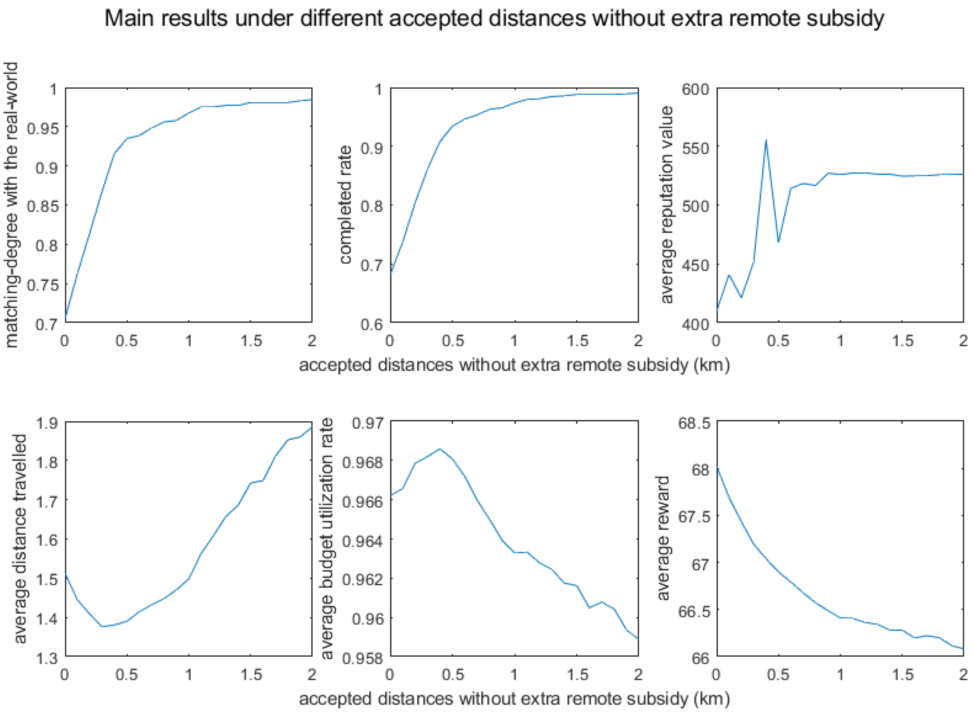

Fig. 5 Main results under different γ values

In comparison with the provided original budget for tasks, we calculate the new budget for the tasks by multiplying a changing amplitude varying from 90% to 110% in 1% increments in the third experiment. We obtain the main results under different budgets as shown in Fig. 6.

If the new budget is lower than the original budget, the allocated rate becomes lower so that only a few tasks can be allocated. The number of allocated tasks is so low that the other indicators under the low allocated tasks can have good performance. If the new budget is larger than the original budget, then the task allocation rate increases slightly and the process of the performance of other indicators is also very unremarkable. Thus, the original budget provided by one Chinese company is very suitable. Hence, for all spatial crowdsourcing tasks, it is very important for task requesters to set a suitable budget.

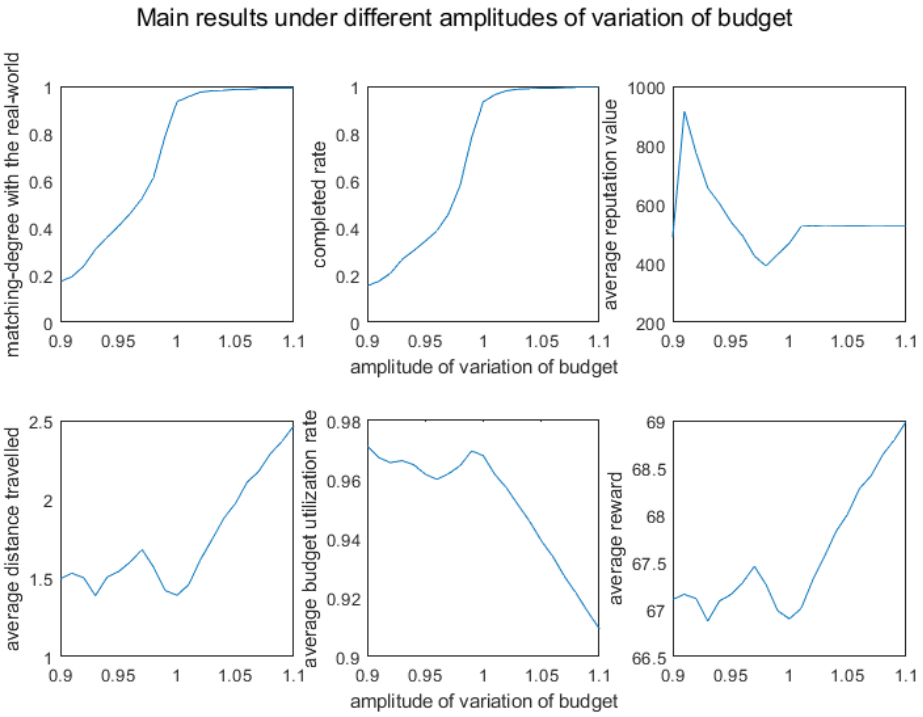

Fig. 6 Main results under different B_j values

5. Discussions and Conclusions

This study focuses on the problem of efficient task package allocation, which is rooted in many real-world applications. Existing methods used to allocate spatial crowdsourcing tasks mainly aim to guarantee the expected quality of information from workers, and lack the consideration of the task allocation rate, which is very low sometimes in the real world [23].

To fill these gaps, we proposed the RB-TPSC with the dual objectives of improving the task allocation rate and guaranteeing the expected quality of information from workers under limited budgets. The proposed RB-TPSC is divided into two stages: the first one maximizing the expected quality of information from workers under limited budgets, and second one aiming to improve the task allocation rate by focusing on those remaining tasks in the first stage. Once the tasks have been allocated to suitable workers, the rewards for these tasks are also determined automatically.

To evaluate the proposed RB-TPSC, we compare it with the Budget-TASC [19]. The results demonstrate that the RB-TPSC significantly outperforms Budget-TASC by over 266.45% in terms of an increase in the average reputation value of workers and in terms of an 8.61% increase in savings on the average distance travelled. Apart from optimizing the expected quality of results from workers, the RB-TPSC remarkably improves the task allocation rate of spatial crowdsourcing tasks from 62.51% to 93.41%.

Moreover, extensive simulation works have been conducted by varying the parameter setting to understand the performance of the proposed RB-TPSC. The simulation results also verify that the proposed RB-TPSC can be implemented as an automatic decision support mechanism to enable task requesters to automatically allocate spatial crowdsourcing tasks to suitable workers.

6. Limitations and further research directions

Although we have considered various characteristics of workers and tasks in the proposed RB-TPSC, some limitations still exist. First, given the insufficient data about the timetables of workers and tasks, we cannot consider the duration of tasks and time requirement of workers, which also influence the specific allocation for spatial crowdsourcing tasks. In our future study, we will try to obtain detailed data and design a realistic algorithm to allocate tasks by considering the specific time characteristics of workers and tasks.

We understand that the Euclidean distance between a requester and a worker is inaccurate to represent the actual traveled distance. We should use the actual distance in our algorithms when the platform can obtain the actual traveled distance with the aid of advanced technologies. However, in this experimental evaluation, we only used the existing longitude and latitude information for calculating the Euclidean distance to represent the actual distances. Further, the distance between the two parties may not be a good measure of travel time as well. One of the primary reasons is that at different period of time, there will be different traffic conditions. Such conditions may increase travel time.

The allocation of spatial crowdsourcing tasks is humane if the willingness of users is considered each time. Currently, real platforms such as Didi Chuxing allocate tasks to suitable users, but the users cannot proactively select the tasks they prefer. In this case, the worker cannot select the tasks but is assigned to the task with the shortest distance if he/she is the top choice of two crowdsourcing tasks. The worker who aims for considerable monetary incentives may prefer a slightly long distance task with high compensation. The current algorithms cannot accommodate such preference. Furthermore, we believe that platforms can utilize technologies to enable the users to write their requirements about their expected tasks at any time and consider their expectations to allocate the tasks.

Acknowledgments

The authors are grateful for the constructive comments of the referees on an earlier version of this paper. Pengkun Wu was supported in part by the Natural Science Foundation of China (71771066 and 71531013) and Joint PhD Programmes (PolyU-HIT) leading to Dual Awards. Ngai was supported in part by The Hong Kong Polytechnic University under grant number ZVK9.

References

[1] S. Erat, V. Krishnan, Managing delegated search over design spaces, Management Science, 58 (3) (2012) 606-623.

[2] B.L. Bayus, Crowdsourcing new product ideas over time: An analysis of the dell ideastorm community, Management Science, 59 (1) (2013) 226-244.

[3] C.-M. Chiu, T.-P. Liang, E. Turban, What can crowdsourcing do for decision support?, Decision Support Systems, 65 (2014) 40-49.

[4] L.B. Jeppesen, K.R. Lakhani, Marginality and problem-solving effectiveness in broadcast search, Organization Science, 21 (5) (2010) 1016-1033.

[5] D. Nevo, J. Kotlarsky, Primary vendor capabilities in a mediated outsourcing model: Can it service

providers leverage crowdsourcing?, Decision Support Systems, 65 (2014) 17-27.

[6] L. Chen, J.R. Marsden, Z. Zhang, Theory and analysis of company-sponsored value co-creation, Journal of Management Information Systems, 29 (2) (2012) 141-172.

[7] H.W. Chesbrough, Why companies should have open business models, MIT Sloan Management Review, 48 (2) (2007) 22-28.

[8] H. Wang, S. Guo, J. Cao, M. Guo, Melody: A long-term dynamic quality-aware incentive mechanism for crowdsourcing, IEEE Transactions on Parallel and Distributed Systems, (2017). 901 - 914.

[9] L. Bradesko, M. Witbrock, J. Starc, Z. Herga, M. Grobelnik, D. Mladenic, Curious cat-mobile, context-aware conversational crowdsourcing knowledge acquisition, ACM Transactions on Information Systems, 35 (4) (2017) 1-46.

[10] Y. Wang, Respecting user privacy in mobile crowdsourcing, Proceedings of the IEEE International Conference on E-Science Workshop, (2013), 50-64.

[11] S. Lazarova-Molnar, H.T. Logason, P.G. Andersen, M.B. Kjaergaard, Mobile crowdsourcing of occupant feedback in smart buildings, Applied Computing Review, 17 (1) (2017) 5-14.

[12] J.B. Pick, O. Turetken, A. Deokar, A. Sarkar, Location analytics and decision support: Reflections on recent avancementa, a research framework and the path ahead, Decision Support Systems, 99 (2017) 1-8.

[13] Y. Huang, A. Shema, H.C. Xia, A proposed genome of mobile and situated crowdsourcing and its design implications for encouraging contributions, International Journal of Human-Computer Studies, 102 (SI) (2017) 69-80.

[14] J. Gonsalves, S. Hosio, M. Vukovic, S. Konomi, Mobile and situated crowdsourcing, International Journal of Human-Computer Studies, 102 (SI) (2017) 1-3.

[15] Y.J. Chang, G. Paruthi, H.Y. Wu, H.Y. Lin, M.W. Newman, An investigation of using mobile and situated crowdsourcing to collect annotated travel activity data in real-word settings, International Journal of Human-

Computer Studies, 102 (SI) (2017) 81-102.

[16] E. Niforatos, A. Vourvopoulos, M. Langheinrich, Understanding the potential of human-machine crowdsourcing for weather data, International Journal of Human-Computer Studies, 102 (SI) (2017) 54-68.

[17] Y. Huang, C. White, H. Xia, Y. Wang, A computational cognitive modeling approach to understand and design mobile crowdsourcing for campus safety reporting, International Journal of Human-Computer Studies, 102 (SI) (2017) 27-40.

[18] T. Sasao, S. Konomi, V. Kostakos, K. Kuribayashi, J. Goncalves, Community reminder: Participatory contextual reminder environments for local communities, International Journal of Human-Computer Studies, 102 (SI) (2017) 41-53.

[19] C.Y. Miao, H. Yu, Z.Q. Shen, C. Leung, Balancing quality and budget considerations in mobile crowdsourcing, Decision Support Systems, 90 (2016) 56-64.

[20] B. Barann, D. Beverungen, O. Müller, An open-data approach for quantifying the potential of taxi ridesharing, Decision Support Systems, 99 (2017) 86-95.

[21] F. Alt, A.S. Shirazi, A. Schmidt, U. Kramer, Z. Nawaz, Location-based crowdsourcing: Extending crowdsourcing to the real world, Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries, (2010), 13-22.

[22] L. Kazemi, C. Shahabi, L. Chen, Geotrucrowd: Trustworthy query answering with spatial crowdsourcing, Proceedings of the 21st ACM Sigspatial International Conference on Advances in Geographic Information Systems, (2013), 314-323.

[23] China society for industrial and applied mathematics, The data of allocated sharing tasks, http://www.mcm.edu.cn/html_cn/node/460baf68ab0ed0e1e557a0c79b1c4648.html, (2017).

[24] B. Guo, H.H. Chen, Z.W. Yu, W.Q. Nan, X. Xie, D.Q. Zhang, X.S. Zhou, Taskme: Toward a dynamic and quality-enhanced mechanism for mobile crowd sensing, International Journal of Human-Computer Studies,

102 (SI) (2017) 14-26.

[25] R.V. Kozinets, K. de Valck, A.C. Wojnicki, S.J.S. Wilner, Networked narratives: Understanding word-of-mouth marketing in online communities, Journal of Marketing, 74 (2) (2010) 71-89.

[26] T. Ludwig, C. Kotthaus, C. Reuter, S. van Dongen, V. Pipek, Situated crowdsourcing during disasters: Managing the tasks of spontaneous volunteers through public displays, International Journal of Human-Computer Studies, 102 (SI) (2017) 103-121.

[27] M.K. Poetz, M. Schreier, The value of crowdsourcing: Can users really compete with professionals in generating new product ideas?, Journal of Product Innovation Management, 29 (2) (2012) 245-256.

[28] J. Bockstedt, C. Druehl, A. Mishra, Heterogeneous submission behavior and its implications for success in innovation contests with public submissions, Production and Operations Management, 25 (7) (2016) 1157-1176.

[29] C. Terwiesch, Y. Xu, Innovation contests, open innovation, and multiagent problem solving, Management Science, 54 (9) (2008) 1529-1543.

[30] J.O. Wooten, K.T. Ulrich, Idea generation and the role of feedback: Evidence from field experiments with innovation tournaments, Production and Operations Management, 26 (1) (2017) 80-99.

[31] K.J. Boudreau, N. Lacetera, K.R. Lakhani, Incentives and problem uncertainty in innovation contests: An empirical analysis, Management Science, 57 (5) (2011) 843-863.

[32] J. Ren, J.V. Nickerson, W. Mason, Y. Sakamoto, B. Graber, Increasing the crowd's capacity to create: How alternative generation affects the diversity, relevance and effectiveness of generated ads, Decision Support Systems, 65 (2014) 28-39.

[33] D. Geiger, M. Schader, Personalized task recommendation in crowdsourcing information systems—current state of the art, Decision Support Systems, 65 (2014) 3-16.

[34] D.E. O'Leary, On the relationship between number of votes and sentiment in crowdsourcing ideas and

comments for innovation: A case study of canada's digital compass, Decision Support Systems, 88 (2016) 28-37.

[35] Y.Y. Han, H.Y. Wu, Minimum-cost crowdsourcing with coverage guarantee in mobile opportunistic d2d networks, IEEE Transactions on Mobile Computing, 16 (10) (2017) 2806-2818.

[36] S. Karnouskos, Crowdsourcing information via mobile devices as a migration enabler towards the smartgrid, 2011 IEEE International Conference on Smart Grid Communications (Smartgridcomm), (2011), 67-72.

[37] C. Willing, K. Klemmer, T. Brandt, D. Neumann, Moving in time and space–location intelligence for carsharing decision support, Decision Support Systems, 99 (2017) 75-85.

[38] C. Zhao, S.S. Yang, X.Y. Yang, J.A. McCann, Rapid, user-transparent, and trustworthy device pairing for d2d-enabled mobile crowdsourcing, IEEE Transactions on Mobile Computing, 16 (7) (2017) 2008-2022.

[39] J. Goncalves, S. Hosio, J. Rogstadius, E. Karapanos, V. Kostakos, Motivating participation and improving quality of contribution in ubiquitous crowdsourcing, Computer Networks, 90 (SI) (2015) 34-48.

[40] S. Hosio, J. Goncalves, V. Kostakos, J. Riekki, Crowdsourcing public opinion using urban pervasive technologies: Lessons from real-life experiments in oulu, Policy and Internet, 7 (2) (2015) 203-222.

[41] J. Li, Y.M. Zhu, Y.Q. Hua, J.D. Yu, Crowdsourcing sensing to smartphones: A randomized auction approach, IEEE Transactions on Mobile Computing, 16 (10) (2017) 2764-2777.

[42] L. Chen, P. Goes, J.R. Marsden, Z. Zhang, Design and use of preference markets for evaluation of early stage technologies, Journal of Management Information Systems, 26 (3) (2009) 45-70.

[43] P. Micholia, M. Karaliopoulos, I. Koutsopoulos, L.M. Aiello, G.D.F. Morales, D. Quercia, Incentivizing social media users for mobile crowdsourcing, International Journal of Human-Computer Studies, 102, (2017) 4-13.

[44] L.G. Jaimes, I.J. Vergara-Laurens, A. Raij, A survey of incentive techniques for mobile crowd sensing,

IEEE Internet of Things Journal, 2 (5) (2015) 370-380.

[45] X.L. Zhang, Z. Yang, W. Sun, Y.H. Liu, S.H. Tang, K. Xing, X.F. Mao, Incentives for mobile crowd sensing: A survey, IEEE Communications Surveys and Tutorials, 18 (1) (2016) 54-67.

[46] J.S. Lee, B. Hoh, Dynamic pricing incentive for participatory sensing, Pervasive and Mobile Computing, 6 (6) (2010) 693-708.

[47] S. Reddy, D. Estrin, M. Hansen, M. Srivastava, Examining micro-payments for participatory sensing data collections, Proceedings of the 2010 Acm Conference on Ubiquitous Computing (Ubicomp 2010), (2010), pp. 33-36.

[48] J.P. Rula, F.E. Bustamante, Crowdsensing under (soft) control, Proceedings of the 34th Annual IEEE International Conference on Computer Communications (INFOCOM), (2015), 2236-2244.

[49] X.L. Zhang, Z. Yang, Z.M. Zhou, H.B. Cai, L. Chen, X.Y. Li, Free market of crowdsourcing: Incentive mechanism design for mobile sensing, IEEE Transactions on Parallel and Distributed Systems, 25 (12) (2014) 3190-3200.

[50] Z. Pan, H. Yu, C. Miao, C. Leung, Efficient collaborative crowdsourcing, Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16), (2016), 4248-4249.

[51] E. Kamar, S. Hacker, E. Horvitz, Combining human and machine intelligence in large-scale crowdsourcing, Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, (2012), 467-474.

[52] P. Dai, C.H. Lin, Mausam, D.S. Weld, Pomdp-based control of workflows for crowdsourcing, Artificial Intelligence, 202 (2013), 52-85.

[53] D. Zhao, X.-Y. Li, H. Ma, How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint, Proceedings of the 33rd Annual IEEE International Conference on Computer Communications (INFOCOM), (2014), 1213-1221.

[54] L. Tran-Thanh, M. Venanzi, A. Rogers, N.R. Jennings, Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks, Proceedings of the 2013 International Conference on Autonomous Agents and Multiagent Systems, (2013), 901-908.

[55] C.-J. Ho, J.W. Vaughan, Online task assignment in crowdsourcing markets, AAAI, (2012), 45-51.

[56] J.X. Liu, Y.D. Ji, W.F. Lv, K. Xu, Budget-aware dynamic incentive mechanism in spatial crowdsourcing, Journal of Computer Science and Technology, 32 (5) (2017), 890-904.

[57] L. Kazemi, C. Shahabi, Geocrowd: Enabling query answering with spatial crowdsourcing, Proceedings of the 20th International Conference on Advances in Geographic Information Systems, (2012), 189-198.

[58] J.B. Ni, X.D. Lin, K. Zhang, Y. Yu, Secure and deduplicated spatial crowdsourcing: A fog-based approach, 2016 IEEE Global Communications Conference (Globecom), (2016), 1-6.

[59] Y.X. Tong, J.Y. She, B.L. Ding, L.B. Wang, L. Chen, Online mobile micro-task allocation in spatial crowdsourcing, 2016 32nd IEEE International Conference on Data Engineering (Icde), (2016), pp. 49-60.

[60] L. Tran-Thanh, T.D. Huynh, A. Rosenfeld, S.D. Ramchurn, N.R. Jennings, Budgetfix: Budget limited crowdsourcing for interdependent task allocation with quality guarantees, Proceedings of the 2014 International Conference on Autonomous Agents and Multiagent Systems, (2014), 477-484.

[61] H. To, L. Fan, L. Tran, C. Shahabi, Real-time task assignment in hyperlocal spatial crowdsourcing under budget constraints, 2016 IEEE International Conference on Pervasive Computing and Communications (PerCom), (2016), 1-8.

[62] C. Veness, Calculate distance, bearing and more between latitude/longitude points, Movable Type Scripts, (2010) 2002-2014.

[63] D.R. Karger, S. Oh, D. Shah, Budget-optimal task allocation for reliable crowdsourcing systems, Operations Research, 62 (1) (2014) 1-24.

[64] H. Yu, C. Miao, B. An, Z. Shen, C. Leung, Reputation-aware task allocation for human trustees,

Proceedings of the 2014 International Conference on Autonomous Agents and Multiagent Systems, (2014), 357-364.