



# Spatial crowdsourcing: a survey

Yongxin Tong<sup>1</sup> · Zimu Zhou<sup>2</sup> · Yuxiang Zeng<sup>3</sup> · Lei Chen<sup>3</sup> · Cyrus Shahabi<sup>4</sup>

Received: 31 December 2018 / Revised: 1 June 2019 / Accepted: 19 August 2019  
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

## Abstract

Crowdsourcing is a computing paradigm where humans are actively involved in a computing task, especially for tasks that are intrinsically easier for humans than for computers. Spatial crowdsourcing is an increasing popular category of crowdsourcing in the era of mobile Internet and sharing economy, where tasks are spatiotemporal and must be completed at a specific location and time. In fact, spatial crowdsourcing has stimulated a series of recent industrial successes including sharing economy for urban services (Uber and Gigwalk) and spatiotemporal data collection (OpenStreetMap and Waze). This survey dives deep into the challenges and techniques brought by the unique characteristics of spatial crowdsourcing. Particularly, we identify four core algorithmic issues in spatial crowdsourcing: (1) task assignment, (2) quality control, (3) incentive mechanism design, and (4) privacy protection. We conduct a comprehensive and systematic review of existing research on the aforementioned four issues. We also analyze representative spatial crowdsourcing applications and explain how they are enabled by these four technical issues. Finally, we discuss open questions that need to be addressed for future spatial crowdsourcing research and applications.

**Keywords** Spatial crowdsourcing · Task assignment · Quality control · Incentive mechanism · Privacy protection

---

✉ Lei Chen  
leichen@cse.ust.hk

Yongxin Tong  
yxtong@buaa.edu.cn

Zimu Zhou  
zzhou@tik.ee.ethz.ch

Yuxiang Zeng  
yzengal@cse.ust.hk

Cyrus Shahabi  
shahabi@usc.edu

- <sup>1</sup> State Key Laboratory of Software Development Environment, Beijing Advanced Innovation Center for Big Data and Brain Computing and International Research Institute for Multidisciplinary Science, Beihang University, Beijing, China
- <sup>2</sup> Computer Engineering and Networks Laboratory, ETH Zurich, Zurich, Switzerland
- <sup>3</sup> Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong SAR, China
- <sup>4</sup> Department of Computer Science, University of Southern California, Los Angeles, CA, USA

## 1 Introduction

Crowdsourcing is a computing paradigm where humans actively or passively participate in the procedure of computing, especially for tasks that are intrinsically easier for humans than for computers. It has attracted extensive attention from both the academia and the industry [59,69,99,141], and there have been many successful crowdsourcing platforms such as Amazon Mechanical Turk (MTurk) [2] and Upwork [28].

With the development of mobile Internet and sharing economy, traditional Web-based crowdsourcing has shifted to spatial crowdsourcing<sup>1</sup> (a.k.a. mobile crowdsourcing) [57, 132,206]. As with traditional crowdsourcing, spatial crowdsourcing involves three components, *tasks*, *workers* and *the platform*. Figure 1 shows the typical workflow of spatial crowdsourcing. The roles of these components are as follows.

- **Tasks** Tasks with spatiotemporal constraints (e.g., the positions and deadlines of tasks) are submitted to the platform. To complete a task, a worker has to physically move to the position of the task.

---

<sup>1</sup> The term was coined for the first time in [132].

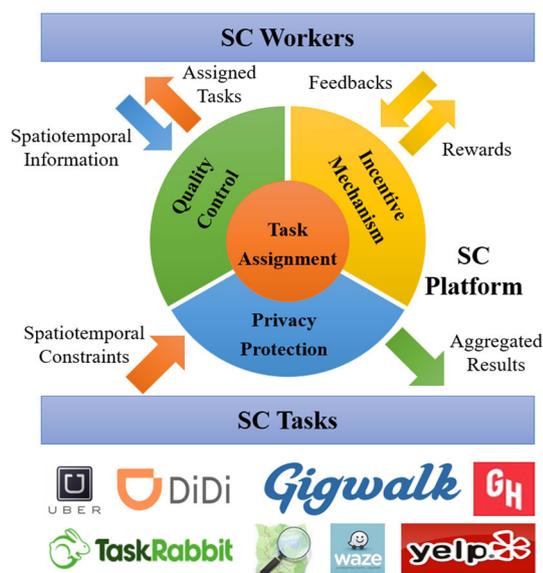


Fig. 1 Key components and workflow in spatial crowdsourcing

- **Workers** Workers submit their spatiotemporal information such as their positions and deadlines to the platform. Depending on the concrete applications, workers either are assigned to tasks or can choose tasks by themselves.
- **The platform** The spatial crowdsourcing platform (platform for short) connects tasks and workers. Its core functions include assigning tasks to suitable works, aggregating the results submitted by workers, setting rewards for workers, and protecting the privacy of the tasks and workers.

The major difference between spatial crowdsourcing and Web-based crowdsourcing is that the former requires each worker to move in the physical world to perform tasks [132]. Hence spatiotemporal information such as location, mobility and the associated contexts plays a crucial role. Its natural connection with the physical world makes spatial crowdsourcing a computing paradigm for a wide spectrum of daily applications including real-time ride-hailing services, e.g., Uber [17] and Didi Chuxing [4], product placement checking supermarkets, e.g., Gigwalk [8] and TaskRabbit [15], on-wheel meal-ordering services, e.g., GrubHub [10] and Meituan [26], and citizen sensing services, e.g., OpenStreetMap [13] and Waze [19].<sup>2</sup>

The emphasis on spatiotemporal dynamics calls for new designs in crowdsourcing theories and systems. The aim of this survey is to provide a comprehensive review on the core

<sup>2</sup> Sometimes Waze is also viewed as a crowdsensing application, which leverages users' sensor-equipped mobile devices to collect and share data. Spatial crowdsourcing is a general framework and can subsume crowdsensing or participatory sensing [132].

algorithmic issues in spatial crowdsourcing from the perspective of the platform.

**Task assignment** In practice, a spatial crowdsourcing platform needs to manage massive tasks and workers every day. For example, in 2017, Didi Chuxing needs to serve 25 million ride requests every day with the registered over 21 million drivers, which eventually produces over 70TB spatiotemporal data every day [21]. Thus, the first challenge of the spatial crowdsourcing platforms is how to assign the large-scale tasks to their workers, i.e., task assignment. The platforms usually aims to arrange the tasks to suitable workers with different optimization objectives such as maximizing the total number of assigned tasks or the total payoff of the tasks to their assigned workers, minimizing the total travel cost of the allocated workers.

**Quality control** As with most crowdsourcing applications, results collected from workers in spatial crowdsourcing vary in quality. The aim of quality control is to quantify the quality of workers and tasks and effectively aggregate results to ensure high-quality task completion. Both the quality models and the aggregation techniques are tied to spatiotemporal information, which imposes unique challenges.

**Incentive mechanism** Proper Incentive mechanisms help attract workers to participate in spatial crowdsourcing. Dedicated incentive mechanism design is needed because the spatiotemporal factors and the relative relation between supply and demand in spatial crowdsourcing are dynamic. For example, if there are only a few workers in some area, the tasks posted in this area should provide more reward.

**Privacy protection** Privacy protection is particularly crucial in spatial crowdsourcing. Spatiotemporal information of workers, tasks, and intermediate results needs to be properly transformed to avoid privacy leakage while allowing efficient information processing such as task assignment. Dedicated techniques and frameworks need to be designed to balance between the strength of privacy protection and the efficiency of other spatial crowdsourcing operations.

**Contributions over existing surveys** There are some general surveys [34,69,99,141] or tutorials [58,59,142] on traditional Web-based crowdsourcing. Our survey focuses on the spatiotemporal factors and the new algorithmic designs on crowdsourcing due to these factors. There are also some surveys or tutorials on spatial crowdsourcing. For example, Guo et al. [104] and Tong et al. [203] review task allocation of spatial crowdsourcing; To et al. review privacy protection of spatial crowdsourcing in Chapter 7 of [195]; Zhang et al. [239] review the incentive mechanisms in spatial crowdsourcing; Zhao et al. [244] give a brief survey on spatial crowdsourcing, which only sketches out a few representative works. Compared with [104,195,239,244], we provide a comprehensive and holistic review on the latest progress

**Table 1** A timeline of milestone papers of spatial crowdsourcing

Year	Reference	Influence
2012	[132]	First work of spatial crowdsourcing
2013	[78]	First work of static task matching (see Sect. 3.3) in spatial crowdsourcing
2013	[133]	First work of quality control (see Sect. 4) in spatial crowdsourcing
2014	[196]	First work of privacy protection (see Sect. 6) in spatial crowdsourcing
2014	[63]	First work of general spatial crowdsourcing platform
2015	[144]	First work of dynamic task planning (see Sect. 3.6) in spatial crowdsourcing
2016	[205]	First work of dynamic task matching (see Sect. 3.4) in spatial crowdsourcing
2016	[204]	First experimental work of dynamic task matching in spatial crowdsourcing
2018	[210]	First work of incentive mechanism (see Sect. 5) in spatial crowdsourcing
2018	[201]	First work of privacy protection in dynamic scenario

on spatial crowdsourcing research. Chen et al. [57] also conduct a survey on spatial crowdsourcing. However, our work is more systematic in classifying the techniques and also covers the most recent literature in the last 3 years. Tong et al. give a tutorial on spatial crowdsourcing in [206]. This survey is its holistic and systematic extension and update.

**Bibliography methodology** We select papers primarily from top venues in the database communities such as SIGMOD, VLDB, ICDE and TKDE. We also include some representative works from the spatial and mobile computing communities since some important algorithmic issues in spatial crowdsourcing also stemmed from there (although as the topic of crowdsensing, which has a slightly different focus). In Table 1, we list the milestone papers during the development of spatial crowdsourcing and their influence on this research area.

In the rest of this survey, we first present the preliminaries in Sect. 2 and review the representative research on the four core issues in spatial crowdsourcing in Sects. 3–6. We then study some killer applications of spatial crowdsourcing in Sect. 7 and discuss future challenges and opportunities in Sect. 8. Finally, we conclude in Sect. 9.

## 2 Preliminaries

This section introduces the models of tasks, the models of workers, and the practical constraints that will be frequently used in this survey.

## 2.1 Task modeling

In spatial crowdsourcing, a task is also known as a spatial task [132], a crowdsourced task [97], a spatial crowdsourced task [96], or a request [154]. The user, who submits the task on such platforms, is called task requester [186] or requester [132]. In real-world applications, a task can be a taxi calling request in ride-sharing platform (e.g., Uber [17] and Didi Chuxing [4]), a takeout order in food delivery platform (e.g., GrubHub [10] and Seamless [27]), a last-mile delivery request in urban logistic platform (e.g., UPS [18] and FedEx [6]), and other general tasks like taking photos of landmarks and appliance repairment in Gigwalk [8] and TaskRabbit [15]. For example, the number of food delivery orders has been increased to 10 billion in China by the end of 2017 [226]. The main reason is that crowdsourcing these tasks can usually result in higher quality task completion (e.g., low latency) at a lower cost due to the large scale of workers.

After receiving the task issued by the requester, the platform will know the following major information about this task.

- **Arrival time** indicates when the task is submitted.
- **Location** represents the spatial information of the task. Some task (e.g., a taxi calling request or a food delivery order) contains two types of locations, **origin** (pickup location) and **destination** (delivery location). To complete such a task, a worker needs to first come to the origin and then take to the destination.
- **Deadline** represents the expired time of the task.
- **Radius** restricts a circular range whose center is the location of the task.
- **Reward** is the payoff to the worker if he/she completes the task. The amount of reward is either directly decided by the requester or determined by the platform based on its incentive mechanisms.

A few other attributes of tasks are also considered in some studies, e.g., required skills [66] (the requirement of skills to perform the task), arrival rate [84] (the probability of appearance in a unit time), etc.

Similar to crowdsourcing [141], the tasks in spatial crowdsourcing can be also classified into two kinds in terms of granularity: i.e., *macro-tasks* and *micro-tasks*. A macro-task in spatial crowdsourcing often involves a wider space and requires more time to complete. In contrast, a micro-task in spatial crowdsourcing usually involves much fewer locations and needs less time to complete. For example, mapping a city belongs to a macro-task whereas geotagging a landmark of this city is a micro-task. As most existing studies in spatial crowdsourcing focus on micro-tasks, this survey also mainly restricts to the scope of micro-tasks and only briefly discusses

**Table 2** A summary about the attributes of tasks and workers used in the four core issues

	Task					Worker				
	Arrival time	Location	Deadline	Radius	Reward	Arrival time	Location	Deadline	Radius	Capacity
Task assignment	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Quality control	✓	✓	✓	✗	✗	✓	✓	✗	✓	✓
Incentive mechanism	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Privacy protection	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗

the issues of task assignment, quality control and incentive mechanism in macro-tasks.

## 2.2 Worker modeling

In spatial crowdsourcing, a worker is also known as a spatial worker [33], a crowd worker [205], a mobile worker [112], a service provider [204], or an agent [55]. To join the platform and perform tasks, a worker usually shares his/her spatiotemporal information with the platform. The commonly used attributes include:

- **Arrival time** indicates when the worker appears on the platform.
- **Location** is the spatial information of the worker.
- **Deadline** restricts the leaving time of the worker.
- **Radius** represents a circular range whose center is the location of the worker.
- **Capacity** is the maximum number of tasks that he/she can perform before the deadline.

From historical data, the platform will also know the acceptance ratio of the worker [237] (the percentage of accepted ones among all the assigned tasks) and the reputation of the worker [108,218]. Some works also consider a few other attributes of workers, e.g., his/her skills [98,187], travel budget [97,245], etc.

Table 2 summarizes the attributes of tasks and workers used in the four core issues in spatial crowdsourcing that we will discuss in the subsequent sections.

## 2.3 Practical constraints

The main characteristic of spatial crowdsourcing is the *spatial* factors (e.g., location) and *temporal* factors (e.g., deadline). These factors are important when the platform makes task assignment, controls the quality, designs the incentive mechanism, and protects the privacy. Thus, existing works usually consider three types of constraints to satisfy the dynamics in spatial crowdsourcing, i.e., **spatial constraints**, **temporal constraints** and **other constraints**. We list the major ones as follows.

### Spatial constraints

- **Range constraint**: the task assigned to a worker is within his/her restricted range; the worker assigned to a task is within its restricted range.
- **Travel budget constraint**: the total travel cost of the worker should be under his/her travel budget.

### Temporal constraints

- **Deadline constraint**: the task will expire after the corresponding deadline; the worker will leave the platform after his/her deadline.
- **Real-time constraint** (a.k.a. instantaneous constraint): once a task appears, a worker must be assigned to it before the next task appears.

### Other constraints

- **Capacity constraint**: the number of tasks assigned to a worker cannot exceed his/her capacity.
- **Invariable constraint**: once a task is assigned to a worker, the allocation between the task and the worker cannot be changed.
- **Reward budget constraint**: the total payoff to the assigned workers should be under the reward budget of the requester.
- **Skill constraint**: each required skills of a task is covered by the skills of at least one worker.
- **Reliability constraint**: the probability of the task being performed correctly should be larger than a threshold of reliability.

Some of these constraints are more widely used in all four core issues, e.g., range constraint, deadline constraint and capacity constraint. Some other constraints are only used in specific scenario, e.g., skill constraint is often used when a task has a specific requirement about the skills of the assigned workers.

### 3 Task assignment

Task assignment is considered as the most fundamental challenge in spatial crowdsourcing [57,206]. This is because all the other core issues in spatial crowdsourcing are connected with task assignment, as we will discuss in the next several sections. In this section, we first define the task assignment problem (Sect. 3.1) and then categorize existing research from two dimensions (Sect. 3.2): the arrival of input (i.e., static or dynamic) and the algorithmic assignment model (i.e., matching or planning). Accordingly, we introduce existing research from four categories: static matching (Sect. 3.3), dynamic matching (Sect. 3.4), static planning (Sect. 3.5), and dynamic planning (Sect. 3.6). Finally, we summarize these studies in Sect. 3.7.

#### 3.1 Generic definition

**Task assignment** aims to arrange tasks to suitable workers for different objectives. A generic definition of task assignment in spatial crowdsourcing is as follows.

*Given a set of tasks and a set of workers, task assignment refers to the process to make an arrangement between tasks and workers for specific objectives, while satisfying spatial constraints, temporal constraints and (or) other constraints.*

In terms of objectives, there are mainly two types of optimization goals: **total utility** and **total cost**. Below are the general definitions of **utility** and the **cost**.

- **Utility** *It is a value that measures the utility of an assignment between a task and its assigned worker. Utility can be a constant value of 1, the reward of the task, the acceptance ratio of the worker, or even the reward times the acceptance ratio. Accordingly, the **total utility** will represent the total number of performed tasks [78], the total acceptance ratio of the workers [237], the total rewards of the assigned tasks [242] or the total expected rewards of the assigned tasks [205,221].*
- **Cost** *It is a value to measure the cost of the assignment between a task and its assigned worker. Cost can be the travel distance (time) of the worker to the task, or the delay of the task from its arrival time to the completion time. Accordingly, the **total cost** will represent the total travel distance of the workers [204] or the total delay of all the tasks [64].*

*Stable* assignment is another objective for certain spatial crowdsourcing applications [224,242]. It is motivated by *stable marriage* [94], which aims to minimize the number of *unstable pairs* (a.k.a. blocking pairs). A pair of task and worker (except its current assigned worker) is an unstable pair if the following conditions are satisfied: (1) The task prefers the worker more than its current assigned worker; (2)

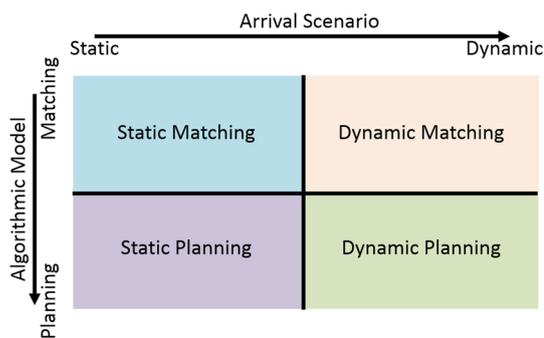


Fig. 2 Matrix overview of task assignment research

the worker prefers the task more than his/her current assigned task.

#### 3.2 Categories of existing research

Figure 2 summarizes the taxonomy of task assignment research. Existing studies can be categorized from two dimensions: the **arrival scenario**, which can be *static* or *dynamic*; and the **algorithmic model**, which can be *matching* or *planning*.

##### Arrival scenario: static versus dynamic

- **Static** In the static scenario (a.k.a. offline scenario), the platform is assumed to know all the spatiotemporal information of the tasks and workers at the beginning, which includes the arrival times and locations of tasks and workers.
- **Dynamic** In the dynamic scenario (a.k.a. online scenario), the spatiotemporal information of either tasks or workers is only known upon their arrival.

Intuitively, the dynamic scenario is more practical yet more challenging than the static one, since tasks and workers in the dynamic scenario need to be assigned based on only partial information.

##### Algorithmic model: matching versus planning

- **Matching** In the matching model, task assignment is often formulated as a bipartite graph-based problem. Workers and tasks can be represented by the vertices in the bipartite graph and utility or cost between a worker and a task can be denoted by the weight of the edges. Then the problem is to obtain an optimal matching in the bipartite graph.
- **Planning** In the planning model (a.k.a. scheduling model), task assignment aims to plan a route for each worker to perform a sequence of tasks.

Before introducing solutions to each category of task assignment research, we list the evaluation metrics for a task assignment algorithm. In terms of *efficiency* of the algorithm, time and memory costs are used. In terms of *effectiveness*, **approximation ratio** and **competitive ratio** are standard to assess the theoretical guarantee of the offline algorithms and online algorithms, respectively. Specifically, the approximation ratio represent the effectiveness that an *offline* algorithm can guarantee in the worst case. The competitive ratio represents the effectiveness that an *online* algorithm can guarantee but under various analysis models. In the task assignment research in spatial crowdsourcing, the following analysis models are covered:

- **Adversarial order model (AO)** It considers the worst case of the online algorithm.
- **Random order model (RO)** It considers the average case of the online algorithm, i.e., the arrival order of inputs is uniformly sampled from all possible permutations.
- **I.I.D model (IID)** It assumes that the dynamically arrived vertices (i.e., workers or tasks) are identical and independent distributed but unknown to the algorithms.
- **Known I.I.D model (KIID)** It also assumes that workers or tasks are i.i.d., but the algorithm knows the distribution.
- **known adversarial distribution model (KAD)** It is a generalization of the KIID model. However, each task and worker in this model is sampled according to an arbitrary distribution, which is known to the algorithm. Different from KIID model, the distributions of the tasks (workers) may be different from each other.

Now we review existing studies in four categories: static matching (Sect. 3.3), dynamic matching (Sect. 3.4), static planning (Sect. 3.5), and dynamic planning (Sect. 3.6).

### 3.3 Static matching

This subsection reviews research on task matching in the static scenario, where information of workers and tasks is known before assignment. We discuss existing studies in terms of different objectives, which include utility maximization (Sect. 3.3.1), cost minimization (Sect. 3.3.2), and stable matching (Sect. 3.3.3).

#### 3.3.1 Utility maximization

In practice, utility can represent the constant value 1 (i.e., the number of assigned task) or the payoff to the worker. Accordingly, the objective of utility maximization is equivalent to either maximizing *the total number of assignments* or *the total payoff*. We discuss existing solutions to these two objectives separately.

**Maximizing total number of assignments** Solutions to static matching problem that maximizes the number of assigned tasks are either *exact* or *Greedy-based* approximate algorithms.

- **Exact** Since task matching can be formulated as a bipartite graph, the maximum cardinality bipartite matching of the graph yields the assignment with the maximum total number. Hence exact algorithms (e.g., Hungarian algorithm [52]) can optimally solve the problem. Alternatively, Kazemi et al. [132] reduce the bipartite graph into an instance of the *maximum flow* problem [32] and use the Ford–Fulkerson algorithm [93] to obtain the exact result. They also consider some practical issues. For example, a task may have fewer workers around and hence it should be assigned with higher priority. Therefore, the authors borrow the idea of location entropy [74] to represent this priority. Location entropy measures the total number of workers near that location as well as the relative proportion of their future visits to that location. Another heuristic strategy is to iteratively assign the task to its nearest worker (i.e., Nearest Neighbor Priority (NNP)).
- **Greedy based** To reduce the computation cost of exact solutions (i.e., Hungarian [52] and Ford–Fulkerson [93] algorithms), various Greedy-based methods are proposed. Both [199] and [212] maximize the total number while considering a budget constraint. They extend the idea of location entropy [132] to region entropy, i.e., tasks in the spatial region with fewer workers inside (i.e., less region entropy) should have a higher priority to be assigned. Then they greedily make assignment based on the current highest priority. Alfarrarjeh et al. [33] further design several partition-based distributed implementations (e.g., spatial partitioning approach (SPA)) to improve the scalability of the solutions.

**Maximizing total payoff** Solutions to static matching that maximizes the total payoff are also either *exact* or *Greedy-based* approximate algorithms.

- **Exact** To et al. [197] extend the problem in [132] by assuming that a worker with better performance should be paid more. Accordingly, an instance of static matching can be reduced to an instance of maximum weighted bipartite matching [52]. Thus the Hungarian algorithm [52] can still be utilized to obtain an exact solution. Considering worker distribution and travel cost, they also reduce the original matching problem to the minimum-cost maximum weighted bipartite matching problem.
- **Greedy based** Similarly, the Greedy-based algorithms are proposed to improve the efficiency of exact algorithms such as the Hungarian algorithm. She et al. [183]

**Table 3** Comparison of existing solutions to task assignment as a static matching problem

Method	Objective	Constraints <sup>a</sup>	Time complexity <sup>b</sup>	Ratio
GR [132]	Maximizing total number	Deadline, range	–	Optimal
SP-WR-A [33]		Range	–	Heuristic
Temporal [199]		Deadline, range, budget	–	Heuristic
Greedy-GEACC [182]	Maximizing total payoff	Capacity	$O(n^3)$	$1/(1 + C_{\max})$
g-D&C [66]		Deadline, range, skill, budget	–	Heuristic
ADAPTIVE [66]		Deadline, range, skill, budget	–	Heuristic
Basic [197]		Deadline, range	–	Optimal
IDA [213]	Minimizing total distance	–	–	Optimal
CA [213]		–	–	Heuristic
Allocation [48]		Capacity	$O(n^3)$	2.5
Swap Chain [156]	Minimizing maximum distance	Capacity	$O(R \cdot  T ( T  +  W ))$	Optimal
Gale–Shapley [94]	Minimizing #blocking pair	Capacity	$O( T  W )$	Optimal
Closest Pair [72,228]		Capacity	$O( T  W ^2)$	Optimal
Chain [224]		Capacity	$O(( T  +  W ) \cdot (\log^{O(1)}  T  + \log^{O(1)}  W ))$	Optimal

<sup>a</sup>In the column of constraints, we use “–” to represent that the method supports no aforementioned constraints in Sect. 2.3. We use “range” to denote range constraints, “deadline” to denote deadline constraints (see Sect. 2.3 for more details)

<sup>b</sup>In the column of time complexity, we use “–” to represent the case when time complexity is not given in the paper. We use  $T$  and  $W$  to denote the set of tasks and the set of workers, respectively. Hence,  $n$  is  $\max\{|T|, |W|\}$  and  $R$  is a parameter such that  $R \ll |T||W|$

consider the conflicts among tasks and propose an approximation solution with ratio  $\frac{1}{1+C_{\max}}$ , where  $C_{\max}$  is the maximum capacity of workers. Cheng et al. [66] study the settings where a task has requirements on the worker’s skills under a budget constraint. Assuming a batch mode [199], they propose a Greedy-based method and further develop a new algorithm with an adaptive cost model.

### 3.3.2 Cost minimization

Since the platform tends to serve more tasks, the actual objective is often to find a matching with maximum cardinality and minimum cost. Hence the corresponding static matching problem can be transferred to the *minimum-cost maximum-flow (MCMF)* problem [32]. The problem can again be solved by exact algorithms such as the Hungarian algorithm [52] and successive shortest path algorithm (SSPA) [81]. To improve the efficiency, Hou et al. [213] leverage indexing and I/O optimization techniques. Specifically, they introduce incremental SSPA-based exact solutions with R-tree indexing. A heuristic algorithm is also designed to achieve better efficiency with an approximate result.

Bei et al. [48] study static matching with cost minimization where each worker can be assigned to at most two tasks at any time. The problem is formulated as a variant of the three-dimensional matching (3DM) [101]. A matching in 3DM is a triple which consists of two tasks and one worker. To solve the problem, they first pack every two tasks and then make an arrangement between the workers and the pairs of tasks after

packing. This two-phase algorithm achieves an approximation ratio of 2.5 when the number of tasks is exactly twice of the number of workers.

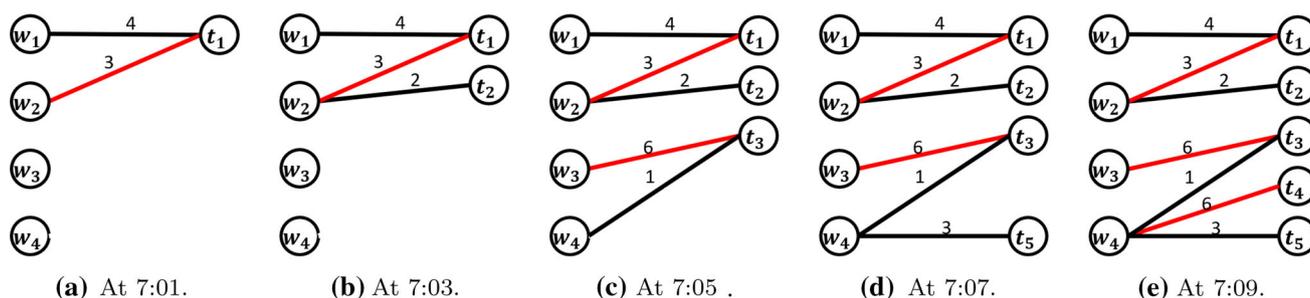
Long et al. [156] also focuses on the static matching with cost minimization. Differently, they want to find a matching with maximum cardinality which minimizes the maximum travel cost among all the assignments. They devise a scalable algorithm called swap chain to efficiently get the optimal solution.

### 3.3.3 Stable matching

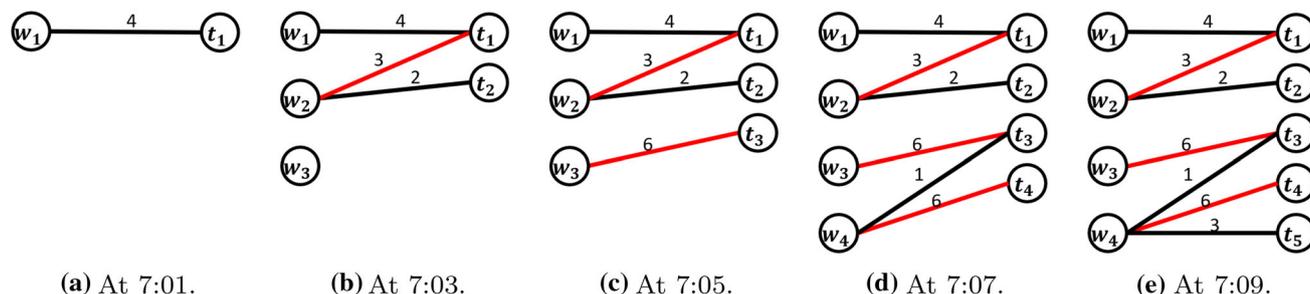
Some other research studies the static stable matching problem in spatial data. Solutions to the stable marriage problem can be applied to static stable matching. For example, the problem can be solved by the Gale–Shapley algorithm [94], which takes  $O(|T||W|)$  time when  $T$  is the set of tasks and  $W$  is the set of workers. Another intuitive solution is to iteratively select the closest pair from the remaining tasks and workers [72,228]. To improve the efficiency, Wong et al. [224] reduce the concept of “mutual nearest neighbor” to the bichromatic mutual NN search problem, and propose an NN search-based chain algorithm with a time complexity of  $O((|T| + |W|) \cdot (\log^{O(1)} |T| + \log^{O(1)} |W|))$ .

### 3.3.4 Summary on static matching

Table 3 lists the representative studies on static matching. Solutions to the static matching problem are the basis of many other more complex task assignment problems in spatial



**Fig. 3** An example of one-sided dynamic matching, where only tasks on the right appear dynamically. The information of workers on the left is known in advance



**Fig. 4** An example of two-sided dynamic matching, where both workers and tasks appear dynamically and the assignment is made immediately

crowdsourcing. Cheng et al. [67] conduct a comprehensive evaluation on mainstream static matching algorithms. They conduct the experiments on the open datasets [9] collected by the gMission and the toolbox called SCAWG [198] for spatial crowdsourcing. According to their experimental results, LLEP [197] is a good choice to maximize the total utility and NNP [132] is closely effective but more efficient.

Besides, solutions to the static matching problem can be also extended for the dynamic scenario via the batch-based mode [132,199]. In the batch mode, an arrangement is made between tasks and workers in a fixed time interval (i.e., a batch). However, the response time of tasks or workers tends to be long in the batch mode. Hence, online algorithms directly designed for dynamic matching are desired, which we present below.

### 3.4 Dynamic matching

This subsection reviews task matching research in the dynamic scenario, where the information of either tasks or workers is unknown beforehand. Dynamic matching can be further classified into *one-sided dynamic matching* and *two-sided dynamic matching*. In one-sided dynamic matching, only the information of workers or tasks is unknown (e.g., parcel delivery), while in two-sided dynamic matching, the information of both workers and tasks is unknown (e.g., on-demand taxi dispatching). Figures 3 and 4 show the examples of one-sided and two-sided dynamic matching, respectively. As in static matching, we review prior works based on their

objectives: *utility maximization* (Sect. 3.4.1), *cost minimization* (Sect. 3.4.2) and *stable matching* (Sect. 3.4.3).

#### 3.4.1 Utility maximization

As in static matching, the utility between a task and its assigned worker in dynamic matching can also represent a constant value 1 and the payoff of the task. Since some real-world applications of dynamic matching allow workers to decide whether to accept the assigned task or not, the utility in dynamic matching can additionally represent the accepted ratio of the worker and further the payoff times the acceptance ratio, i.e., expected payoff. Accordingly, the main objectives for dynamic matching with utility maximization include *maximizing the (expected) total number of assigned tasks* and *maximizing the (expected) total payoff of assigned tasks*.

**Maximizing total number of assignments** Dynamic matching with this objective is also known as the online bipartite matching problem. Most research along this line focuses on the *one-sided online bipartite matching* problem [50,82,91,102,121,129,159], while relatively a few have investigated *two-sided online bipartite matching* [120,208,220].

(i) **Solutions to one-sided scenario** We discuss solutions optimized for worst-case performance and average performance, respectively.

- **Optimized for worst-case performance** Karp et al. [129] propose three algorithms, i.e., GREEDY, RANDOM and

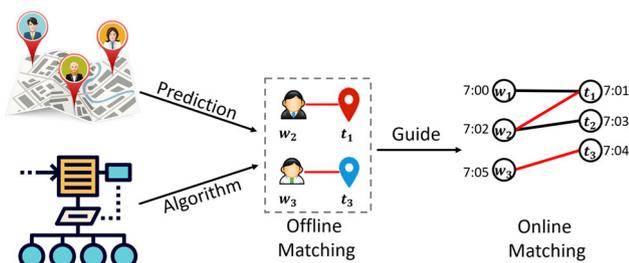


Fig. 5 Procedure of offline-guide-online technique

RANKING. GREEDY assigns a new task to an arbitrarily chosen available worker. RANDOM differs in that the worker is uniformly sampled. The competitive ratios of GREEDY and RANDOM are both 0.5 under the adversarial order model. RANKING is a two-phase algorithm. In the first phase, a random permutation of workers is picked and it represents the priority (i.e., rank) of the workers. In the second phase, a newly appeared task will be assigned to the available worker with the highest rank. RANKING yields a competitive ratio  $1 - 1/e \sim 0.632$  under the adversarial order model. The ratio is proven to be the lower bound of any online algorithm [50,82,102].

- **Optimized for average performance** While RANKING has achieved the lower bound under the adversarial order model, it is unknown whether it is the most effective under other models such as the random order model and the i.i.d model [102]. Feldman et al. [91] devise the *suggested matching* algorithm with a higher ratio 0.67. The idea is to guide the online algorithm by offline solutions (“offline-guide-online”). Figure 5 shows the procedure of the offline-guide-online technique. Specifically, they first predict the spatiotemporal information of tasks and workers (i.e., learn the distribution). Next, an offline matching algorithm is used to obtain the optimal matching by using the predicted inputs. Finally, they use the offline matching to guide the online matching policy. When a new task appears, an eligible worker is usually sampled according to the chosen probability of this assignment in the offline solution. Following the same idea, Manshadi et al. [159] improve the ratio to 0.702 using Monte Carlo sampling. Jaillet et al. [121] apply linear program as the offline algorithm and obtain the best-known competitive ratio of 0.706.

(ii) **Solutions to two-sided scenario.** Many solutions to the two-sided scenario are built upon those to the one-sided scenario.

GREEDY can be applied to two-sided scenario and achieves a competitive ratio of 0.5 under the adversarial order model. Its effectiveness can be further improved by two meth-

ods, the randomized primal dual technique [120,220] and the offline-guide-online technique [208].

In particular, the charging-based framework [128] can be extended to achieve better effectiveness. Its idea is to increase the probability of each potential assignment whenever a worker or a task appears on the platform. At the deadline of each vertex, the algorithm will determine the final assignment of this vertex based on the probability. Wang et al. [220] extend the framework with the water-filling algorithm and obtain a better ratio of 0.526 than GREEDY under adversarial order model. Huang et al. [120] extend RANKING into the two-sided scenario where the vertex with higher rank has a larger probability to be matched. The extended RANKING algorithm achieves the currently best-known competitive ratio of 0.554.

Tong et al. [208] apply the *offline-guide-online* technique to a different setting where a worker can move in advance to other locations so as to increase the potential number of assignments. Their solution first predicts the spatiotemporal information of tasks and workers and guides workers to locations where there will be tasks in the future, and then makes assignments based on an offline guide. The proposed POLAR and POLAR-OP algorithms yield competitive ratios of 0.399 and 0.47 under the i.i.d model.

**Maximizing expected total number of assignments** When workers are allowed to reject the assigned tasks, the objective above is replaced by maximizing the expected total number of assigned tasks.

Hassan et al. [111] use multi-armed bandit [176] to model the problem and apply a contextual bandit algorithm [143] to determine the assignments. Zhang et al. [237] focus on predicting the acceptance ratio of workers in taxi dispatching via machine learning techniques. Note that tasks rejected by workers can be considered as new tasks and can be reassigned to other workers.

**Maximizing total payoff** Dynamic matching that maximizes the total payoff can be considered as an *online vertex-weighted bipartite matching* problem, where the weight of each edge in bipartite graph is represented by the weight of one-side vertex. There are also two versions of this problem, i.e., one-sided [31,51,194] and two-sided [84,194].

(i) **Solutions to one-sided scenario** In [31], Aggarwal et al. study the problem where the information of tasks is known. A perturbed Greedy algorithm is proposed which achieves a competitive ratio  $1 - 1/e \sim 0.632$  under adversarial order model. Specifically, the algorithm first perturbs each weight of vertices identically and independently by a function  $\psi(x) = 1 - e^{-(1-x)}$ . Then it sorts the vertices in the order of decreasing perturbed weights, which forms a rank. Finally, it utilizes the strategy of RANKING [129] to make the final decision. The authors prove that no randomized algorithm can obtain a higher ratio than 0.632 under the adversarial order model. Ting et al. [194] devise a random-

ized algorithm Greedy-RT to achieve the ratio of  $\frac{1}{2e \lceil \log U_{\max} + 1 \rceil}$  under the adversarial order model, where  $U_{\max}$  denotes the upper bound of the utility between a worker and a task. They first randomly sample a threshold and then match the new vertex to any existing vertex whose weight is higher than the threshold. Under the known i.i.d model, Brubach et al. [51] propose the VW algorithm with a competitive ratio of 0.729.

(ii) **Solutions to two-sided scenario** For the two-sided scenario, Ting et al. [194] prove that Greedy-RT can still achieve the ratio of  $\frac{1}{2e \lceil \log U_{\max} + 1 \rceil}$  under the adversarial order model. They also prove that no randomized algorithm can achieve a higher ratio than  $\frac{2}{\lceil \log U_{\max} \rceil + 1}$  under the adversarial order model. Dickerson et al. [84] design an algorithm ADAP based on the offline-guide-online technique. They first solve a linear program benchmark and then use the offline solution to simulate the online matching procedure. Finally, they prove that the competitive ratio of ADAP is 0.343 under the random order model.

**Maximizing expected total payoff** Similar to the case of maximizing the expected total number of assignments, this thread of research assumes the worker can reject the task with some probability. In this case, the weight of edges in the bipartite graph is determined by both the payoff of the task and the acceptance ratio of the worker. Hence the problem is similar to the **online edge-weighted bipartite matching** problem. Again, there are two versions of this problem, i.e., one-sided [51,83,134,139] and two-sided [84,205].

(i) **Solutions to one-sided scenario** Prior works either borrow the idea from the secretary problem [92] or use the offline-guide-online technique [51,83].

Korula et al. [139] first propose a Sample-And-Price algorithm that has a competitive ratio of 0.125 under the random order model. The idea is to iteratively find a global assignment by GREEDY whenever a new vertex appears, and then sample an assignment with some probability. Kesselheim et al. [134] are also motivated by the secretary problem and devise the BOM algorithm, which improves the competitive ratio to  $1/e \sim 0.367$  under the random order model. Different from Sample-And-Price, BOM skips the first  $\lfloor (|W|+|T|)/e \rfloor$  vertices and finds a global optimal matching by the Hungarian method.

Solutions which use the offline-guide-online technique to obtain more promising results. In [51,83], the authors first formulate the predicted instance with linear programming (LP), and then solve it by existing LP solver (e.g., CPLEX [20]). They finally use the result of the LP solver to guide the online matching procedure. Under the known i.i.d model, the SW algorithm [51] can achieve a competitive ratio of 0.632 and an optimized algorithm EW [51] can obtain a ratio of 0.705. In [83], the authors exploit the fact that a worker tends to be reassigned a new task right after he/she finishes the last task, and define the online matching with (offline) reusable resources problem.

They propose a Monte Carlo simulation-based algorithm  $ADAP(\gamma)$ , which achieves a competitive ratio of 0.5 under the known adversarial distribution model.

(ii) **Solutions to two-sided scenario** In [205], the authors extend the Greedy-RT algorithm [194] and prove its competitive ratio still holds. They also borrow the idea of secretary problem and devise a two-phase framework. In the first half of vertices, GREEDY is used to determine the final assignment. In the other half of vertices, they first find a global matching and then determine the final assignment based on the global matching. Their proposed algorithms achieve competitive ratios of 0.25 and 0.125 under the random order model. Dickerson et al. [84] further use the offline-guide-online technique to improve the ratio to 0.295. Song et al. [186] study a variant of the problem for applications such as InterestingSport [11] and Nanguache [12], where workplaces, workers and tasks should all be considered. For example, InterestingSport needs to find suitable trainers (i.e., workers) and book the corresponding sports facilities (i.e., workplaces) for its users. The problem is modeled as online trichromatic matching. A threshold-based randomized framework is proposed to solve the problem with a ratio of  $\frac{1}{3e \lceil \log U_{\max} + 1 \rceil}$ .

**Summary** Most of the efforts on dynamic matching with utility maximization can be modeled as a variant of online bipartite matching problem. The offline-guide-online technique [91] is useful to achieve good competitive ratios, e.g., [84,208]. However, the common assumption is that the spatiotemporal distribution of either tasks or workers is completely predictable, which may be impractical in real-world applications.

### 3.4.2 Cost minimization

The cost between a task and a worker can represent the travel distance (time) between the location of the worker and the location of the task, or the delay of the task from release time to completion time. Hence cost minimization indicates that tasks will be served more rapidly. We discuss solutions that minimize the *total travel distance* and the *total delay* separately.

**Minimizing total travel distance** dynamic matching with this objective can be modeled as a variant of one-sided online minimum bipartite matching, where tasks dynamically appear on the platform. Existing solutions can be classified into two categories, *Greedy* [127] and *HST based* [46, 161] algorithms.

- **Greedy based** Kalyanasundaram et al. [127] propose Permutation, a  $(2n - 1)$ -competitive ratio algorithm, where  $n$  is the number of workers to be matched. They also introduce Greedy, which greedily assigns a task to its closest worker (i.e., nearest neighbor) and randomly picks one if there is a tie. Despite its efficiency, Greedy

has a competitive ratio of  $2^n - 1$  under the adversarial order model. In order to distinguish from the GREEDY algorithm [129] on utility maximization, we call this nearest-neighbor-based Greedy method as NN-Greedy.

- **HST based** Hierarchically separated tree (HST) [223] is a special type of tree metrics. Meyerson et al. [161] consider a randomized Greedy algorithm, HST-Greedy, by extending the Permutation algorithm [127] into HST structures, and it yields an expected competitive ratio of  $O(\log^3 n)$  on any metric space. The  $O(\log^3 n)$  bound is further improved in [46] by the HST-Reassignment algorithm, which is  $O(\log n)$ -competitive on 2-HST metrics, and thus  $O(\log^2 n)$ -competitive on general metrics.

The above studies focus on analyzing the competitive ratios in worst cases i.e., under the adversarial order model. To evaluate the performance of these algorithms in practice, Tong et al. [204] present a comprehensive experimental comparison of some representative algorithms. The experiments show that the NN-Greedy, which has always been considered as the worst method due to its exponential competitive ratio ( $2^n - 1$ ), significantly outperforms the others in terms of effectiveness. In particular, the worst case in the adversarial order model of NN-Greedy has a constant competitive ratio, 3.195 under the random order model.

**Minimizing total delay** The delay of a task is the duration from its release time to its completion time. Unlike minimizing the total travel distance, minimizing the total delay is a group of problems where once a task appears, it can be kept waiting for potential better assignments instead of being matched immediately. The cost incurred is the sum of travel distances between matched worker–task pairs (the travel cost), and the sum of the tasks’ response time (the waiting cost). The rationale is to trade off between the cost of instant assignments and that of waiting for better assignments.

- **Solutions to one-sided scenario** Emek et al. [88] present a randomized algorithm with competitive ratio  $O(\log^2 n + \log \Delta) \sim O(\log^2 n)$  on  $n$ -point metric spaces with the longest distance  $\Delta$ . Ashlagi et al. [42] prove the same ratio with a simpler analysis and Azar et al. [43] further improve the ratio to  $O(\log n)$  under the adversarial order model.
- **Solutions to two-sided scenario** Chen et al. [64] study the problem of minimizing the maximum delay among all matches while both tasks and workers dynamically appear. They present an HST-based algorithm MMD-HST, which has better effectiveness than a Greedy-based baseline.

**Summary** Dynamic matching with cost minimization is usually modeled as an online minimum bipartite matching

problem. There are mainly two kinds of solutions to this problem, Greedy-based and HST-based algorithms. Compared with Greedy-based algorithms, HST-based algorithms tend to have better competitive ratios in worst-case analysis. However, since NN-Greedy is demonstrated to be effective on both synthetic and real datasets [204], it is still an open problem whether the competitive ratio of NN-Greedy is arbitrarily bad (i.e.,  $2^n - 1$ ) under other analysis models (e.g., random order model or known i.i.d model).

### 3.4.3 Stable matching

Many studies on dynamic matching have integrated the preferences of *either* workers *or* tasks into their optimization objectives. For example, some studies [83,205] aim to maximize the total utility obtained from all successful assignments, where the utility represents the workers’ preference on payoff.

Zhao et al. [242] first consider the preferences of both workers and tasks in dynamic task matching and formulate the task assignment problem as a variant of online stable matching problem. The online stable matching problem is first studied by Khuller et al. [135]. They prove that the “first come, first served” method (FCFS-Greedy) produces  $O(n \log n)$  blocking pairs on average and  $O(n^2)$  blocking pairs in worst case. Zhao et al. [242] study a more difficult version since they also aim to maximize the total utility at the same time. They use the offline-guide-online technique [91] and propose an LP-based algorithm LP-ALG, which achieves a competitive ratio of  $1 - 1/e \sim 0.632$  for maximizing total utility with no more than  $0.6|E|$  blocking pairs under the known i.i.d model.

### 3.4.4 Summary on dynamic matching

Table 4 compares the representative research on dynamic matching with three objectives (utility maximization, cost minimization and stable matching). As is shown, the competitive ratio is often constant in solutions to utility maximization and different analysis models are often used to obtain a better result. However, fewer studies focus on minimizing the total cost or online stable matching. In particular, most research [46,88,161] uses the adversarial order model to analyze the effectiveness of the algorithm in the worst case, which is more difficult to obtain a promising result. Thus, it is still an open problem whether it is possible to design an algorithm with a constant competitive ratio under the random order model or the known i.i.d model. Finally, it is worth mentioning that many research (e.g., [145,208,242]) conducts the experiments of dynamic matching on the datasets collected by Didi Chuxing [4]. Didi Chuxing has so far already released many open datasets in their GAIA initiative [22]. These real

**Table 4** Comparison of existing solutions to task assignment as a dynamic matching problem

Method	Objective	Constraints	Time complexity <sup>a</sup>	Analysis model <sup>b</sup>	Ratio	
GREEDY [129]	Maximizing total number	One-sided, range	$O(n^2)$	AO	0.5	
GREEDY [102]		One-sided, range	$O(n^2)$	RO, IID	0.632	
RANKING [129]		One-sided, range	$O(n^2)$	AO	0.632	
Suggested matching [91]		One-sided, range	$O(n)$	IID	0.632	
MC sampling [159]		One-sided, range	$O(n)$	IID	0.702	
Random Lists [121]		One-sided, range	$O(n^2)$	IID	0.706	
GREEDY [120]		Two-sided, range, deadline	$O(n^2)$	AO	0.5	
Water-filling [220]		Two-sided, range	$O(n^2)$	AO	0.526	
Ext-RANKING [120]		Two-sided, range, deadline	$O(n^2)$	AO	0.554	
POLAR-OP [208]		Two-sided, range, deadline	$O(n^2)$	IID	0.47	
Contextual bandit [111]	Maximizing expected total payoff	Two-sided, range, deadline	$O(n^2)$	–	Heuristic	
Hill-climbing [237]		Two-sided, range	$O(n^3)$	–	Heuristic	
Perturbed Greedy [31]		One-sided, range	$O(n \log n)$	AO	0.632	
VW [51]		One-sided, range	$O(n^2)$	KIID	0.729	
Greedy-vRT [194]		Two-sided, range	$O(n^2)$	AO	$\frac{1}{2e \ln(U_{\max} + 1)}$	
ADAP [84]		Two-sided, range, deadline	$O(n^2)$	KIID	0.343	
Sample-And-Price [139]		One-sided, range	$O(n^3)$	RO	0.125	
BOM [134]		One-sided, range	$O(n^4)$	RO	0.367	
SW,EW [51]		One-sided, range	$O(n^2)$	KIID	0.632, 0.705	
ADAP( $\gamma$ ) [84]		Two-sided, range	$O(n)$	KAD	0.5	
TGOA [205]	Minimizing total distance	Two-sided, range, deadline	$O(n^4)$	RO	0.25	
TGOA-Greedy [205]		Two-sided, range, deadline	$O(n^3 \log n)$	RO	0.125	
NADAP [84]		Two-sided, range, deadline	$O(n)$	KIID	0.295	
Permutation [127]		One-sided	$O(n^3)$	AO	$2n - 1$	
NN-Greedy [127]		One-sided	$O(n)$	AO	$2^n - 1$	
HST-Greedy [161]		One-sided	$O(n)$	AO	$O(\log^3 n)$	
HST-Reassignment [46]		One-sided	$O(n^2)$	AO	$O(\log^2 n)$	
Stilt-walker [88]		Minimizing total delay	One-sided	–	AO	$O(\log^2 n)$
Saturated [43]			One-sided	–	AO	$O(\log n)$
TGM [41]			One-sided	–	AO	$O(\log n)$
MMD-HST [64]	Two-sided		–	AO	$O(\log n)$	
FCFS-Greedy [135]	Minimizing #blocking pair and minimizing #blocking pair	One-sided	$O(n^2)$	AO	$O( E )$	
LP-ALG [242]		One-sided, range	$O(n)$	KIID	0.632 and $0.6 E $	

<sup>a</sup>In the column of constraints, we use “range” to denote range constraints, “deadline” to denote deadline constraints (see Sect. 2.3 for more details)

<sup>b</sup>In the column of time complexity, we use “–” to represent that the time complexity is not given in the paper. We use  $n$  to denote the maximum value between the number of tasks and the number of workers

<sup>c</sup>In the column of analysis model, we use “–” to represent that the paper has no competitive analysis under specific models

datasets can usually be used to validate the performances of the dynamic matching algorithms for different objectives.

### 3.5 Static planning

Task assignment in the real applications such as ride sharing and food delivery is a planning problem, where a route (i.e., a sequence of tasks) should be planned for workers. This subsection reviews studies on static planning, which fall into two categories, **One-Worker-To-Many-Tasks Static Planning** (Sect. 3.5.1), which plans a route for one single worker, and **Many-Workers-To-Many-Tasks Static Planning** (Sect. 3.5.2), which plans routes for multiple workers.

#### 3.5.1 One worker to many tasks

In One-Worker-To-Many-Tasks Static Planning, most studies aim to find a route for one worker such that the number of performed tasks is maximized under the travel budget constraint. This problem is closely related to the orienteering problem [214]. The major differences include: (1) the utility value of each matching is often zero or one, and (2) the end vertex of the route is not given. Thus, the utility often represents a constant value 1 in the majority of works [78,80] and only [73] considers the more general utility (i.e., payoff). We discuss existing works based on their objectives.

**Maximizing total number of assignments** Deng et al. [78] first study static planning which maximizes the total number of performed tasks under the travel budget and deadline constraints. They name it the Maximum Task Scheduling (MTS) problem and prove its NP-hardness. There are two kinds of solutions to the this problem: *exact* and *Greedy-based* algorithms.

- **Exact** To address the MTS problem, Deng et al. [78] propose several exact solutions. They first propose a dynamic programming algorithm, MST-DP, with a time complexity of  $O(n^2 2^n)$  and a space complexity of  $O(n 2^n)$ . They further propose a branch-and-bound-based algorithm MST-BB, which has a time complexity of  $O(n!)$  and a space complexity of  $O(n^2)$ . They also propose several pruning strategies to improve the actual running time.
- **Greedy based** Deng et al. [78] also propose several Greedy-based heuristics, including nearest-neighbor heuristic (NNH), most promising heuristic (MPH) and least expiration time heuristic (LEH). Among these solutions, NNH is the most efficient and effective. To achieve a better trade-off between efficiency and effectiveness, they further present beam search heuristic (BSH) [80]. It expands the cardinality of candidate set to a given threshold instead of one in NNH. BSH then invokes MST-BB with this candidate set to select proper tasks. Even though

BSH is less efficient than NNH, it is more effective in experimental evaluations.

**Maximizing total payoff** Costa et al. [73] study static planning which maximizes the total payoff. They assume that a worker may be on his/her preferred path and is willing to consider the trade-off between payoff and the travel cost. Due to its NP-hardness, they propose a detour-oriented heuristic (DOH) to find all non-dominated routes and recommend them to the workers.

#### 3.5.2 Many workers to many tasks

Although it is already NP-hard to plan a route for a single worker, a few efforts have explored Many-Workers-To-Many-Tasks Static Planning. Research on Many-Workers-To-Many-Tasks Static Planning mainly focuses on maximizing the general utility (e.g., satisfaction score [97,181], payoff [113]) while only [79] aims at maximizing the total number of performed tasks.

**Maximizing total number of assignments** deng et al. [79] extend their maximum task scheduling problem to the multiple workers version. They devise a new three-phase framework called global assignment and local scheduling (GALS). The first two phases are static matching and One-Worker-To-Many-Tasks Static Planning. The last phase is to refine the matching result with the updated routing result. The last two phases repeat until no more tasks can be performed. The complexity of GALS is  $O(n^4)$ , which is relatively high in practice. Thus, they propose the local assignment local scheduling (LALS) algorithm based on the similar idea to improve the efficiency.

**Maximizing total payoff/satisfaction** In practice, the utility function can represent the satisfaction score [97,181] between workers and tasks or the payoff of the worker by performing the task [113]. There are mainly two types of solutions to this problem, *greedy based* [97,181] and *local ratio based* [113,181] algorithms.

- **Greedy based** She et al. [181] propose the problem of Utility-aware Social Event-participant Planning (USEP), which maximizes the total satisfaction of all the users considering the travel budget constraint. They propose a Greedy-based algorithm, RatioGreedy, which considers the utility–cost ratio of each worker–task pair and adds the pair with the largest ratio into the planning. Gao et al. [97] study a variant of the problem, where tasks may impose different skill requirements on the workers. They first form a set of worker with minimum cardinality to cover the skill requirement of tasks and then greedily assign the worker with the largest satisfaction to the tasks.
- **Local ratio based** The main idea of the local ratio framework [47] is to first decompose the problem into several

**Table 5** Comparison of existing solutions to task assignment as a static planning problem

Method	Objective	Constraints	Time complexity <sup>a</sup>	Ratio
MST-BB [78]	Maximizing total number	Deadline	$O(n!)$	Optimal
NNH [78]		Deadline	–	Heuristic
GALS [79]		Deadline	$O(n^4)$	Heuristic
LRBA [113]	Maximizing total utility	Travel budget	–	5
DeDPO [181]		Deadline, travel budget	$O(n^3)$	0.5
SCUP [97]		Deadline, travel budget, skill	$O(n^2)$	Heuristic

<sup>a</sup>In the column of constraints, we use “deadline” to denote deadline constraints (see Sect. 2.3 for more details)

<sup>b</sup>In the column of time complexity, we use “–” to represent that the time complexity is not given in the paper. We use  $n$  to denote the maximum value between the number of tasks and the number of workers.

simpler subproblems and then eliminate the conflict of these subproblems. She et al. [181] propose a two-phase algorithm called DeDP. It achieves the approximation ratio of 0.5 with time complexity of  $O(n^3)$  and space complexity of  $O(n^2)$ . To improve the efficiency, they further devise an optimized algorithm DeDPO. To maximize the reward of the performed tasks, He et al. [113] propose a local ratio-based algorithm, LRBA. They also use the same technique to prove that the approximation ratio of LRBA is 5. Experimental results show that LRBA outperforms a Greedy-based algorithm.

### 3.5.3 Summary on static planning

Table 5 summarizes existing works on static planning. Static planning in spatial crowdsourcing has been studied in two settings, One-Worker-To-Many-Tasks Static Planning and Many-Workers-To-Many-Tasks Static Planning. Since One-Worker-To-Many-Tasks Static Planning is NP-hard, the Greedy-based solutions are proposed to improve the efficiency of exact solutions. However, all Greedy-based solutions have no theoretical guarantee in the effectiveness. The local ratio technique is often exploited to design an approximation solution. Experiments [113,181] on the Meetup datasets in [153] show that the local ratio-based algorithm is more effective than the Greedy-based solution.

## 3.6 Dynamic planning

Dynamic planning is the planning problem where the information of workers or tasks is unknown in advance. It is more challenging than static planning since the routes of workers have to be planned when only partial information is available. As with static planning, we review research on dynamic planning in two categories: **One-Worker-To-Many-Tasks Dynamic Planning** (Sect. 3.6.1) and **Many-Workers-To-Many-Tasks Dynamic Planning** (Sect. 3.6.2).

### 3.6.1 One worker to many tasks

Research on One-Worker-To-Many-Tasks Dynamic Planning often maximizes the total utility under the budget of travel cost. As before, we review two kinds of total utilities, *the total number of assignments* [144] and *the total payoff of the workers* [190].

**Maximizing total number of assignments** Li et al. [144] prove that under the adversarial order model, no deterministic algorithm has a constant competitive ratio. They propose several Greedy-based approaches such as nearest-neighbor heuristic (NN-Greedy) and earliest deadline heuristic (ED-Greedy). They further propose a bidirectional search-based algorithm to improve the effectiveness. The search begins with the origin and the destination of the worker. Some pruning strategies are proposed to reduce the searching space.

**Maximizing total payoff** Sun et al. [190] extend the problem in [144] to maximize the total payoff to workers. They devise an NN-greedy-based algorithm to balance three influence factors on a worker’s choice in terms of which task to undertake next. They further borrow the idea of offline-guide-online technique [91] to enhance the effectiveness and efficiency.

### 3.6.2 Many workers to many tasks

Among the planning problems discussed in this survey, dynamic planning for multiple workers is the most challenging. We review existing literature with the objectives to maximize the total number of assignments [38], maximize the total payoff [40,192,247] or minimize the total travel distance [119,158].

**Maximizing total number of assignments** In [38], the authors design an auction-based framework. In the framework, workers give out their bids according to their best schedule if incorporating the new task and the platform then selects a worker for the task.

**Maximizing total payoff** Tao et al. [192] devise two algorithms: delay planning and fast planning to solve the problem.

**Table 6** Comparisons of existing solutions to task assignment as a dynamic planning problem

Method	Objective	Constraints	Time complexity <sup>a</sup>	Analysis model <sup>b</sup>	Ratio
Re-Route [144]	Maximizing total number	Deadline	–	AO	Heuristic
Auction-SC [38]		Deadline	–	–	Heuristic
Fast-Planning [192]	Maximizing total payoff	Deadline	$O(n^3)$	AO	Heuristic
APART [40]		Deadline, budget	–	AO	Heuristic
EPBR [190]		Deadline, range	–	–	Heuristic
PBM [247]		Deadline, budget	$O(n^3)$	–	Heuristic
t-share [158]	Minimizing total travel distance	Deadline	–	–	Heuristic
kinetic [119]		Deadline, budget	–	–	Heuristic
pruneGreedyDP [211]	Minimizing unified cost	Deadline	$O(n^2 + n^2 \log n)$	AO	Heuristic

<sup>a</sup>In the column of constraints, we use “range” to denote range constraints, “deadline” to denote deadline constraints (see Sect. 2.3 for more details)

<sup>b</sup>In the column of time complexity, we use “–” to represent that the time complexity is not given in the paper. We use  $n$  to denote the maximum value between the number of tasks and the number of workers

<sup>c</sup>In the column of analysis model, we use “–” to represent that the paper has no competitive analysis under specific models

In delay planning, the worker, who has not finished his/her currently assigned tasks, will not be allocated to the newly arrived tasks. Instead, the route of a worker in fast planning may be updated when new tasks arrive. Both [40] and [247] focus on maximizing the total payoff in another type of application, ride sharing. Asghari et al. [40] propose a branch-and-bound solution to find the optimal routes. Zheng et al. [247] devise an order matching-based solution.

**Minimizing total travel distance** Both [158] and [119] aim to minimize the total travel distance while trying to serve all requests. Ma et al. [158] first study the dynamic task planning for ride-sharing service on a road network. A filter-and-refine-based framework *t-share* is devised with grid index. Based on a similar framework, Huang et al. [119] design a trie-based data structure called *kinetic tree*. The kinetic tree applies the procedure of *insertion* to update the route of each worker.

### 3.6.3 Summary on dynamic planning

Table 6 compares existing works on dynamic planning. Existing studies on dynamic planning, particularly those for ride sharing service, has two main limitations. First, the optimization objectives in some papers are conflicting (e.g., [158] and [119]). Second, some solutions are inefficient. Specifically, some algorithms are inefficient when the capacity of workers becomes larger. For example, [247] restricts that the capacity is no more than 2 and [119] can not response in real time anymore when the capacity becomes 6 (see experiments in [211]). Major solutions rely on inefficient *insertion* procedure [119,158]. To address these limitations, Tong et al. [211] abstract a unified formulation of dynamic planning in sharing transportation, i.e., URPSM problem, which generalizes the previous two objectives. They further design a novel dynamic programming based *insertion* operation to improve the efficiency. They compare their solutions with the

state-of-the-art algorithms on two large-scale datasets, i.e., the GAIA datasets [22] collected by Didi Chuxing and the NYC datasets [16] collected from the taxis in New York City. Experiments on these two datasets show that their framework *pruneGreedyDP* outperforms *t-share* [158] and *kinetic* [119].

## 3.7 Discussions

We summarize representative studies on each category of task assignment in Table 3 (static matching), Table 4 (dynamic matching), Table 5 (static planning) and Table 6 (dynamic planning). Almost all these papers focus on the micro-tasks rather than macro-tasks. This is because a macro-task (e.g., mapping a city) is usually decomposed into large numbers of micro-tasks (e.g., geotagging a landmark in this city) on real-world platforms. Then the algorithms can still be used to determine the allocation between workers and decomposed micro-tasks. Comparing these studies, many focus on the dynamic scenario instead of the static scenario and there are more papers on matching than planning. It seems that the offline-guide-online technique is helpful to obtain a better competitive ratio in dynamic task matching under the known i.i.d model or the known adversarial distribution model. We also observe that there is no competitive algorithm in dynamic planning. Thus, the offline-guide-online technique from dynamic matching may be a starting point to devise competitive algorithms for dynamic planning. Finally, despite extensive research on either static planning or dynamic planning, there is still no comprehensive evaluation on these solutions either empirically or theoretically.

## 4 Quality control

One characteristic of crowdsourcing is that tasks are performed by workers of diverse quality. Quality control aims

to ensure high-quality task completion in presence of diverse worker quality, which is achieved by allowing *multiple* workers to perform *the same* task. Quality control in traditional crowdsourcing roughly deals with two issues: (1) how to quantify the quality of workers and tasks; and (2) how to aggregate results from workers of diverse qualities to meet the quality requirements of tasks. The spatiotemporal factors add new dimensions in both issues, which we discuss in this section.

## 4.1 Quality modeling

The definition of worker and task quality is application-specific. We focus on the worker and task quality related to spatiotemporal factors.

### 4.1.1 Quality of worker

First we discuss worker quality used in traditional crowdsourcing (*inherent worker quality*) and then the new factors in spatial crowdsourcing (*spatiotemporal related worker quality*). Finally, we briefly review the methods to estimate the quality of workers.

**Inherent worker quality** Worker quality in traditional crowdsourcing can be modeled by worker probability [53, 105,209], confusion matrix [215,222] and diversity of skills [115,246]. Specifically, the worker probability approach uses a single value to model the quality of a worker. The value can be the accuracy, confidence, experience or reputation of the worker. A large value normally means a high worker quality. However, the single-valued quality may not suffice to characterize the worker quality for some complex tasks. Hence multi-dimensional approaches such as vectors and confusion matrices are proposed to describe worker quality. The elements in the vectors or confusion matrices represent various skills of workers and the conditional probabilities with different truth values. For example, a normalized four-dimensional vector  $(0.30, 0.78, 1.00, 0)^T$  may represent a worker's abilities on Java, Python, Ruby and C#. Each row of a confusion matrix is the probability distribution under the condition of different correct answers. In general, the vector and matrix approaches characterize workers in more detail and outperform the single-valued worker probability model [248].

**Spatiotemporal related worker quality** In spatial crowdsourcing, quality of workers is often affected by extra spatiotemporal constraints. For example, in addition to an inherent quality as mentioned above, each worker is also assumed to have a distance-aware quality in crowdsourced POI labeling applications [117]. In fact, it is common for spatial crowdsourcing applications to assume that workers can only reliably perform tasks within a certain range [133,205].

**Assessment of worker quality** The assessment methods of worker quality vary for different aspects of worker qual-

ity. Assessment of the inherent quality is usually based on historical data [63,65,133,237]. For example, the historical accuracy to perform tasks is used to estimate the accuracy of a worker to perform future tasks [65,237]. Spatiotemporal related quality is often set via various spatiotemporal data processing models. For distance-aware quality, parameter estimation methods like Bayesian [100,167,168] and maximum likelihood estimation [117] are adopted to evaluate worker qualities with different distance sensitivities.

### 4.1.2 Quality of task

On the one hand, similar to traditional crowdsourcing, the quality of a crowdsourcing task is evaluated by reliability, which is usually formalized as the probability that over 50% workers correctly answer the task [133,180,219] or the chance that at least one worker successfully completes the task [103,240]. Specifically, [133] was the first work to consider the quality issue in spatial crowdsourcing. These studies [133,180,219] focus on the spatial tasks that needs a qualified answer, e.g., spatial data collection by taking photos. Therefore, the requester of the task usually has an expectation of the final answers. Differently, another type of tasks only needs to be successfully completed by one worker, e.g., the on-demand taxi calling service in Didi Chuxing [4]. Thus, such studies [103,240] focus on the probability that at least one worker can eventually finish the task.

On the other hand, unlike the crowdsourcing tasks commonly seen in traditional crowdsourcing, the spatiotemporal factors may directly reflect the quality of tasks in spatial crowdsourcing.

**Latency as task quality** Latency of tasks is closely related to the quality of service for a spatial crowdsourcing platform. Specifically, Zeng et al. [232] consider the maximum latency of all tasks as a criterion for task quality. This criteria is commonly used in real-world applications like Facebook Editor [5] and OpenStreetMap [13]. Differently, Das et al. [75] consider the average latency of all tasks as a criterion for task quality. The average latency is usually considered as the quality of tasks in taxi-dispatching platform (e.g., Uber [17] and Didi Chuxing [4]) or food/parcel delivery platform (e.g., Meituan [26] and Cainiao [3]).

**Diversity as task quality** Diversity is particularly important for event detection or labeling applications. For example, a POI may need to be labeled multiple times by different workers so that reasonably accurate and complete information about the POI can be obtained [118]. Cheng et al. [65] first consider the diversity in the quality of tasks. They observe two types of diversity from the tasks in spatial crowdsourcing: *spatial diversity* and *temporal diversity*.

Specifically, spatial diversity is important when some tasks ask the workers to take photographs/videos of the city landmarks from different angles. When there are  $r$  workers

around the task, the authors use the entropy to define the spatial diversity (SD) as

$$SD = - \sum_{j=1}^r \frac{A_j}{2\pi} \cdot \log \left( \frac{A_j}{2\pi} \right), \quad (1)$$

where  $A_j$  is the angles between two results (photos).

Temporal diversity is important when some tasks require the workers to complete the tasks at different time intervals. For instance, a vacant parking space needs to be monitored at different time windows [65]. If there are  $r$  workers who will be working at each time interval of the whole working period  $T$ , the temporal diversity is also defined based on the idea of entropy as

$$TD = - \sum_{j=1}^{r+1} \frac{t_j}{T} \cdot \log \left( \frac{t_j}{T} \right), \quad (2)$$

where  $t_j$  is the  $j$ th time interval.

The two kinds of diversity can also be combined to assess the spatiotemporal diversity (STD) of a task:

$$STD = \beta \cdot SD + (1 - \beta) \cdot TD, \quad (3)$$

where  $\beta$  is a parameter to balance the importance of spatial diversity and temporal diversity.

## 4.2 Result aggregation

Given the worker quality and the results from multiple workers, aggregation techniques derive the final result for each task so that the quality requirements of tasks can be satisfied. Typical aggregation techniques [248] include Majority Voting [53,140], Weighted Majority Voting [115,147], Probabilistic Graphical Models [77,174], etc. Aggregation techniques in spatial crowdsourcing need to account for spatiotemporal factors, which brings in new aggregation techniques.

In the task of real-time urban traffic speed estimation, workers are assigned to collect or voluntarily contribute traffic data in different locations, and the goal of the task is to reliably estimate the traffic speed in the road network. For example, in [116,155], the systems recruit workers to probe the real-time traffic speed of some roads, while Waze [19] collects traffic data from users' mobile phones to estimate the average speed when its users drive around with the app turned on. Existing studies generally ignore the quality of workers, implicitly assuming that the data collected by workers are reliable. In addition, it is often the case that limited number of workers can be recruited to measure the traffic speed because of the budget constraint, i.e., only the speeds of part of road segments are available. Therefore,

the problem boils down to choosing the optimal subset of road segments to measure in order to maximize the quality of speed estimation of the entire road network. Hu et al. [116] study the real-time urban traffic speed estimation problem where only the speeds on a predefined number of roads (seeds) can be obtained by spatial crowdsourcing. They propose five algorithms (SUPGREEDY, RANDOM, MAXCOV, COVGREEDY, HYBRIDGREEDY) to select seeds and present a two-step model to estimate the speeds of other roads, taking advantage of the correlation among roads. Specifically, the first step constructs a probability graphical model to infer the traffic trend and the second step estimates the traffic speed using a hierarchical linear model. Evaluations on the taxi datasets [1] collected in Beijing and Nanjing show a traffic speed estimation accuracy around 80%. Similar to [116], Liu et al. [155] capture two statistical properties of speed, periodicity, and correlation, using a probabilistic graphical model. They propose to select the best set of workers to probe the real-time traffic speed for the corresponding roads using a hybrid Greedy-based algorithm with an approximation ratio above  $(1 - \frac{1}{e})/2$ . The traffic speed of the entire road network is then estimated using speed propagation based on the model constructed beforehand. The final false estimation rate of the proposed method on the gMission dataset [63] is around 0.08.

In the crowdsourced POI labeling task, a graphical probability model is proposed to deduce the correct labels [117]. Assuming that the labeling results follow a conditional distribution on worker quality, POI influence and the true labels, the authors propose a maximum likelihood estimation (MLE) and expectation maximization (EM) method to estimate the unknown probability parameters and labeling results.

In the task of crowdsourced event detection, reports from different workers are aggregated to detect the true event [100,168]. In [168], the problem is formulated as truth inference under missing or wrong reports. The authors model missing and wrong reports based on the location popularity, the truth of events and the participant reliability, and propose a recursive inference algorithm to infer the latent variables and the truth of events. The method is extended in [100] by considering the state of event as a function of time. The authors design inference algorithms to update the conditional probability of report and variables recursively until the true label of event converges. The Kalman filter is also used to improve the approximation to the event truth.

In the tasks of collaborative mapping, workers often voluntarily participate in map making without financial compensation. In such applications (e.g., OpenStreetMap [13] and Wikimapia [29]), the major purpose of the macro-task is to map a large region (e.g., city), which can be decomposed into large numbers of micro-tasks (e.g., mapping a landmark). Quality control for such macro-tasks, i.e., obtain-

**Table 7** Comparison of representative studies on quality control in spatial crowdsourcing

Reference	Quality modeling		Aggregated method <sup>a</sup>
	Worker	Task	
[133]	Probability	Reliability	Majority voting
[232]	Probability	Latency and reliability	Majority voting
[65]	Probability	Diversity and reliability	Majority voting
[219]	Probability	Reliability	Weighted majority voting
[103]	Probability	Reliability	–
[240]	Probability	Reliability	–
[100]	Probability	Reliability	Bayesian estimation
[167,168]	Probability	reliability	Bayesian estimation
[117]	Probability and distance	Reliability	Expectation maximization

<sup>a</sup>In the column of aggregated method, we use “–” to represent that the paper aims to guarantee that at least one worker can successfully complete the task (e.g., on-demand taxi dispatching) and hence the proposed method does not need to consider the aggregation

ing the qualified results of the macro-task, consists of three steps.

- **Assessment of worker/task quality** Since the workers are usually volunteered, the qualities of workers and tasks may notably differ in practice [114,179]. On one hand, the inherent quality of worker is usually based on historical records and the user profiles [235]. On the other hand, the quality of task can be evaluated based on spatiotemporal diversity [65,70]. Existing work also uses the densities of the tasks to assess the quality of task [70], e.g., the number of provided answers over the area of the region, the number of volunteered workers over the population of the region, etc.
- **Aggregation of micro-tasks** With the decomposition of the macro-tasks, the results of each micro-tasks can be independently aggregated. Therefore, typical aggregation techniques include voting [248] and rating [179] can be applied. Some platforms like OpenStreetMap [13] also allow the expert workers to help validate the aggregated answers.
- **Removal of inconsistencies** Finally, the results of some micro-tasks may be conflicting from the global view of the macro-task, e.g., administrative boundaries self-intersect or split instead of being closed-loop sequences of roads. Thus, existing work also investigates removing the consistencies between the micro-tasks. Keep-Right [24] is a data consistency check tool for OpenStreetMap which can detect errors in the map data, such as loops, overlapping ways, and missing boundaries. Hashemi et al. [110] present a similarity-based framework to detect the logical, topological inconsistencies according to the spatial relationships of micro-tasks.

A few studies have also explored deep learning [56] in collaborative mapping.

### 4.3 Discussions

In a sense, quality control and task assignment in spatial crowdsourcing are interwoven. Table 7 summarizes existing studies on quality control.

On the one hand, the quality metrics of workers and tasks in Sect. 4.1 can be directly applied as either a constraint or an objective in the task assignment problems in spatial crowdsourcing. For example, in [65], maximizing the expected spatial/temporal diversities and the smallest reliability among all tasks are regarded as part the objective of task assignment. In the maximum correct task assignment problem [133], a correct match between a task and assigned workers should satisfy two spatial constraints: (i) tasks should be in the spatial region of assigned workers; (ii) aggregated reputation of workers should exceed a preset threshold of tasks.

On the other hand, the aggregation techniques in Sect. 4.2 can be combined with effective task assignment to further improve the quality of task completion. For example, in crowdsourced POI labeling, the authors divide the problem into label inference and task assignment [117]. In label inference, the accuracy of a label is determined by worker quality and POI influence. In task assignment, they use MLE to estimate the parameters mentioned above and the final results of labels. Then they adopt a Greedy-based algorithm which selects the assignment with maximum accuracy improvement for current workers. In [116], the speed estimation task is completed in two steps. The first step is task assignment which selects  $K$  roads that can best perform speed estimation. After obtaining the speeds of  $K$  roads, the second step is to infer the speed of other roads based on the these  $K$  roads.

## 5 Incentive mechanism

Any crowdsourcing involves certain incentive mechanisms to attract active and qualified workers. Incentive mechanisms determine the rewards to workers such that more workers can be motivated to perform the tasks. Compared with the incentive mechanisms in traditional crowdsourcing, incentive mechanisms in spatial crowdsourcing not only need to attract the interests of workers (which is similar), but also to involve reliable workers to physically move to the location of tasks (which is unique). Since the locations of workers may change over time, the incentive mechanisms in spatial crowdsourcing also need to account for the spatiotemporal factors. In this section, we first introduce the commonly used evaluation metrics in the design of incentive mechanisms (Sect. 5.1). Next, we divide existing works into two categories: **posted price models** (Sect. 5.2) and **auction-based models** (Sect. 5.3). In posted price models, the platform first determines the reward for workers and workers can only accept it or not. Conversely, in auction-based models, workers can first submit their expected reward and the platform then determines the rewards to the workers afterward. Finally, we compare existing studies in Sect. 5.4.

### 5.1 Evaluation metrics

An incentive mechanism is assessed from two aspects: **algorithm metrics** and **mechanism metrics**.

**Algorithm metrics** In spatial crowdsourcing, an incentive mechanism is often an algorithm. Thus, the common algorithm metrics are also used to assess the efficiency and effectiveness of a mechanism.

- **Complexity** Complexity analysis includes the running time and memory usage of the algorithm, which reflects the efficiency of an incentive mechanism. In particular, the *computational efficiency* of a mechanism represents whether the algorithm can be terminated in polynomial time.
- **Approximation/competitive ratio** Approximation ratio and competitive ratio guarantee how bad an algorithm is compared with the optimal solution in the worst case in the offline scenario and the online scenario, which reflect the effectiveness of an incentive mechanism.

**Mechanism metrics** As a functional mechanism, an incentive mechanism should have the properties below.

- **Truthfulness** A truthful mechanism guarantees that workers always submit the truthful information (e.g., the expected reward based on his/her private evaluation) to the platform. In other words, they cannot obtain more revenue by submitting false information about themselves,

where the revenue of a worker represents his/her reward minus his/her cost to perform the task.

- **Individual rationality** An individually rational mechanism guarantees that each participated worker will obtain a nonnegative revenue, i.e., the reward to the worker is no less than the cost of the worker to perform the task.
- **Budget balance** A budget-balanced mechanism guarantees that the total reward to workers does not exceed a given budget, i.e., the mechanism does not need more budget from outside.

### 5.2 Posted price models

The posted price model is widely used in applications like taxi dispatching (e.g., Uber [17]) and food delivery (e.g., Meituan [26]). In this model, the platform determines the reward to the worker and the worker can only decide whether to accept the task or not. Incentive mechanisms following this model can be further divided into two types, *Supply-and-Demand-Aware Model* and *Quality-Aware Model*. In the first type, the rewards are mainly determined based on the comparison between supply (i.e., the number of workers) and demand (i.e., the number of tasks). In the second type, the rewards are mainly determined based on the quality of workers or the quality of tasks.

#### 5.2.1 Supply-and-demand-aware model

In spatial crowdsourcing applications, the **supply** (i.e., the number of workers) and the **demand** (i.e., the number of tasks) often vary in space and time [207]. The corresponding incentive mechanism should reflect the spatiotemporal dynamics between supply and demand. That is, the reward to the worker and the payment of the requester should be dynamic, i.e., *dynamic pricing*. Compared with the traditional fixed price strategy (i.e., static pricing), the incentive mechanisms based on this model are more likely to obtain higher total revenue, which has already been validated in real-world applications e.g., the *surge pricing* in Uber [17].

In the model, a **base price** represents the long-term price, which is usually determined based on prior knowledge of the markets. According to the dynamics of supply and demand, an incentive mechanism changes the unit reward on basis of the base price or the most recently used price.

A well-known adoption of this model is the *surge pricing* in Uber [17], which has been studied in [54,61,122,138,157]. Specifically, during times of high demand for rides, the unit fare may change by multiplying the base price with a multiplier accordingly to the incentive mechanism of *surge pricing*. Thus, the areas with higher multipliers usually indicate a steady stream of ride requests (i.e., tasks), where drivers (i.e., workers) will be attracted to come to. As a result, this incentive mechanism will eventually ensure that

the pickup is quick and reliable. Experiments show that the surge pricing strategy not only reduces the waiting times of tasks, but also improves rewards for workers [157].

The supply-and-demand-aware model has also attracted extensive academic research.

Banerjee et al. [44] apply queuing theories to analyze the incentive mechanisms in ride sharing. They propose a *single-threshold*-based dynamic pricing, where the unit fare for tasks reduces to a lower value if the number of workers is above the threshold. They find that the *single-threshold* dynamic pricing is robust and can be applied to find an optimal base price.

Both [45] and [60] apply Markov process to determine the fare to tasks and the reward to workers. Banerjee et al. [45] still assume that tasks appear on the platform following the queuing model and their pricing strategy is determined by Markovian transitions between independent state (i.e., the distributions of workers on the platform). They obtain the approximate solution by relaxation techniques. Chen et al. [60] consider more spatiotemporal issues, e.g., travel time and driver direction. They use Markov decision process (MDP) to formulate the problem, i.e., the driver distribution on each vertex of graph as a state, the throughputs of tasks on each edge as actions, and the transitions between states as the revenue. Even though it is PSPACE-hard to solve MDPs, they design an polynomial-time algorithm to find an approximate result.

Differently, Tong et al. [210] use bipartite graphs to model the Global Dynamic Pricing (GDP) problem. They aim to find the optimal pricing strategy along with the task assignment. First, they propose a Myerson Reserve Price-based algorithm to determine the base price for each urban area. Based on this base price, they further propose a matching-based algorithm with an approximation ratio of  $1 - 1/e \sim 0.632$  to dynamically adjust the unit price for each area according to the dynamics of supply and demand.

Other studies [39,90,184] focus on the incentive mechanisms specifically for ride sharing. Fang et al. [90] use *subsidies* to provide incentives to workers such that enough supplies can be ensured. Their experiments show that *subsidies* are effective to avoid supply shortages. Asghari et al. [39] take the future changes of supply and demand into consideration. Their intuition is that in regions where the supply is abundant, lowering the prices can lead to higher demand which in turn increases the number of requests.

Shen et al. [184] integrate the task planning into the design of incentive mechanisms in dynamic scenario. They develop an Integrated Online Ridesharing Mechanism (IORS), which satisfies desirable properties such as truthfulness, individual rationality, and budget balance. Their experiments show that compared to an auction-based mechanism [68] (which we will introduce later), IORS achieves a very close performance with substantially less computational time.

## 5.2.2 Quality-aware model

Sometimes tasks are expected to be accomplished with high quality, especially in applications like crowdsourced spatiotemporal data collection. The quality-aware model takes quality into account when providing incentives to workers. We focus on how to design effective incentives (i.e., determine the reward to attract reliable workers), which is related to, but different from quality control in Sect. 4. According to the types of quality discussed in Sect. 4, we divide incentive mechanisms using quality-aware models into two types, *quality-of-worker-aware* [218,225,230] and *quality-of-task-aware* [151,163]. Note that most of the studies above are under a reward budget constraint, i.e., the total rewards of workers should not exceed the budget of the task.

**Quality-of-worker-aware** Studies of this type consider the reputation of workers [218,230] or the willingness of workers in terms of spatial factors [225] when deciding the reward regarding the quality of workers.

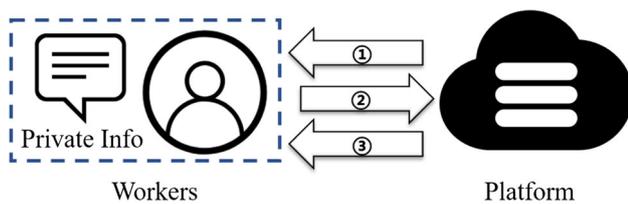
Yu et al. [230] and Wang et al. [218] model the quality of workers with their reputation. They both assume that workers are classified into three kinds: *high* reputation, *medium* reputation or *low* reputation. The rewards of workers are determined by the reputation level, i.e., the worker with higher reputation will obtain higher reward. However, a worker with low reputation will not be paid, since they assume the requester does not like to engage such a worker.

Wu et al. [225] consider the distance between workers and tasks. In general, workers prefer tasks nearby [173]. Therefore, in [225], extra remote subsidies should be paid if workers far away are selected. The subsidy increases linearly with the distance between the worker and the task but no higher than a threshold. The final reward for a worker consists of the base price (calculated with the local average payment per unit time), the subsidy, and the extra tips for more incentives.

**Quality-of-task-aware** Studies in this type consider the latency [163] or the spatial diversity [151] with regard to quality of tasks.

Mitsopoulou et al. [163] try to minimize the latency of tasks by incentive mechanisms. They propose an adaptive pricing policy. Specifically, workers will receive a penalty if they do not respond immediately, i.e., workers providing responses with longer latency will get less reward, and the penalty increases with the latency. The parameters of the reward function can be tuned for every worker. So by adjusting the parameters, the platform can make more workers respond to the tasks, or make workers respond more quickly.

Liu et al. [151] provide incentives to workers in consideration of the spatial diversity. They study the case where there is a task which needs to collect data from different places and propose a price adjustment function. This function allocates more money to the workers doing tasks in such places where



**Fig. 6** Workflow of a basic auction model for incentive mechanism design in spatial crowdsourcing. Step (1): Announcement; Step (2): Bidding; Step (3): Rewarding

data already collected is less than the expected amount. As a result, more workers will be attracted to such places, and the imbalance of data collection among different places can be mitigated. With enough data in each place required by the task, higher quality can be achieved.

Many real applications apply the idea of giving more rewards to remote places for more data, considering the spatial diversity. Pokémon Go [14] is among one of the most successful. In this gamification-based spatial crowdsourcing platform, users can use their mobile phones to track and catch Pokémon (virtual monsters). According to a recent study [202], the platform can also collect the spatial data via GPS. By placing attractive Pokémons at different locations, the platform can stimulate players to go there. As a result, diversified and qualified data can be collected.

### 5.3 Auction-based model

Posted price models determine the reward to workers based on the *estimated* expectation of workers. When the estimation is wrong or unavailable, the reward might be improperly set. Auction-based models overcome this disadvantage by permitting workers to bid the task with their own expectation (i.e., private information including the reward they expect, etc.) and then determining the reward to the worker afterward. We first introduce the workflow of the auction model in Sect. 5.3.1, and then review the representative works in Sect. 5.3.2.

#### 5.3.1 Workflow

Figure 6 illustrates the workflow of a basic auction model. It includes three steps.

- (1) **Announcement** The platform first announces the task to the workers who are possible to complete it under the spatiotemporal constraints.
- (2) **Bidding** After receiving the announcement from the platform, workers bid based on their private information (e.g., submit their expected payment) to the platform. In this step, a worker can be strategic and selfish. Hence he may submit the fake information to earn higher reward (i.e.,

an untruthful worker). The incentive mechanisms should guarantee the truthfulness of the worker.

- (3) **Rewarding** The platform decides the reward according to the collected private information. Besides, the platform sometimes also needs to determine the task assignment along with the reward.

Next we review representative auction-based incentive mechanisms. Since the first step of **announcement** is the same for most mechanisms, we mainly discuss the **bidding** and the **rewarding** procedures of different mechanisms as well as their performances.

#### 5.3.2 Representative auction-based mechanisms

We review auction-based incentive mechanisms for two applications, ride sharing [37,40,236] and citizen sensing services [68,243].

(i) **Auction-based incentives for ride sharing.** Both [40] and [37] focus on incentive mechanisms to maximize the total revenue of the platform, i.e., total payment of requesters minus the total rewards to workers. They both use an auction-based model to determine the reward to the worker along with the assignment of tasks. Specifically, after receiving the announcement from the platform, the worker will locally calculate the updated route which is the most profitable to complete this new task. Next, the worker will bid their expected reward and the calculated route to the platform. Finally, the platform will assign the task to the worker with the highest revenue from this task.

In [37,40], the payment of requester is determined based on the calculated route of its assigned worker. In [40], the authors apply a first-price auction scheme (i.e., to pay the highest reward to the worker with highest bid) to determine the reward to the worker while in [37] the researchers use the second-price auction scheme (i.e., to pay the second highest reward to the worker with highest bid) to determine his/her reward. Since a second-price auction model can guarantee a few promising mechanism properties, the incentive mechanism used in [37], SPARP, is truthful and individually rational. Finally, they conduct the experiments on the real-world datasets of New York City's taxis [16]. Experimental results show that [37] can obtain more total revenue than the first-price auction scheme.

Zhang et al. [236] adopt another auction model, *double auction*. In their incentive mechanism, both workers and requesters provide their bids based on private information (i.e., the expected reward to workers and the expected payment of requester) to the platform at the same time. After receiving the private information from the two sides of the market, the platform will make a task matching between workers and requesters and determine the actual reward and the actual payment. They also design a discounted trade

**Table 8** Comparison of representative studies on incentive mechanisms in spatial crowdsourcing

Incentive model	Reference	Time complexity <sup>a</sup>	Ratio	Truthfulness	Individual rationality	Budget balance
Auction	[40]	–	–	✗	✗	✗
Auction	[37]	–	–	✓	✓	✓
Auction	[236]	–	–	✓	✓	✓
Auction	[68]	–	–	✓	✓	✓
Auction	[238]	$O(\frac{1}{\epsilon} W ^3 \log( T ))$	$O(\frac{1}{\epsilon})$	✓	✓	✗
Auction	[243]	$O( T  W  \min( T ,  W ))$	0.25	✓	✓	✓
Quality-aware	[230]	–	–	✗	✗	✓
Quality-aware	[218]	$O( W )$	–	✗	✗	✓
Quality-aware	[225]	–	–	✗	✗	✓
Quality-aware	[151]	$O( T  \times (\# \text{ of Time Windows}))$	–	✗	✗	✓
Quality-aware	[163]	–	–	✗	✗	✓
Supply-and-demand-aware	[210]	$ G  \log  G  +  P  \times \min( T ,  W )(\log  G  +  E )$	0.632	✗	✗	✗
Supply-and-demand-aware	[44]	–	–	✓	✗	✗
Supply-and-demand-aware	[60]	Polynomial	–	✗	✗	✗
Supply-and-demand-aware	[39]	$O( G ^3)$	–	✗	✗	✗
Supply-and-demand-aware	[90]	–	–	✓	✗	✗

<sup>a</sup>In the column of time complexity, we use “–” to represent that the time complexity or ratio is not given in the paper. We use  $T$  and  $W$  to denote the set of tasks and the set of workers, respectively

<sup>b</sup> $|G|$  is the number of regions,  $|P|$  is the number of discrete prices, and  $|E|$  is the number of possible assigned pairs of tasks and workers

reduction mechanism to make a discount in both actual reward and actual payment, DTR, which is truthful, individually rational, and budget-balanced.

Cheng et al. [68] study the incentive mechanism design in the last-mile delivery service. In the step of bidding, the worker sends his/her direct travel distance and a compensation rate (i.e., the cost in unit distance) to the platform. The authors devise the *bottom-up* mechanism to determine the actual reward along with routing plan. Their mechanism is truthful, individually rational, and budget-balanced.

(ii) **Auction-based incentives for citizen sensing.** Zhang et al. [238] propose an auction-based incentive mechanism in the online scenario. Specifically, each worker dynamically appears on the platform, and proposes his/her bid (including the expected reward) to the platform. The platform assigns the tasks to the selected workers and determines the reward to them, in order to maximize the total utility. Their incentive mechanism TOIM is computationally efficient, truthful, individually rational and profitable (i.e., the platform will get a nonnegative revenue from the mechanism).

Zhao et al. [243] also focus on the incentive mechanism design in the online scenario, with any monotone submodular [166] objective function and a budget constraint. In the bidding step, each worker submits his/her expected reward and the sets of tasks that he/she would like to accomplish. In

the last step, the platform selects a subset of workers based on an adaptive threshold such that the total reward to these workers does not exceed the budget. They propose two incentive mechanisms, OMZ and OMG, to handle the cases when workers immediately leave the platform and the workers can stay for a time period, respectively. The mechanisms are truthful and individually rational with a competitive ratio of 0.25 and time complexity of  $O(mn \min(m, n))$  at each time, where  $m$  and  $n$  are the number of requests and workers. To validate the effectiveness of the proposed mechanisms, they conduct the experiments in the Wi-Fi signal sensing application provided by [185]. The experimental results show that both OMZ and OMG achieve the approximate result as the offline optimal solution. In particular, OMZ is often better than OMG in terms of effectiveness.

## 5.4 Discussions

In summary, an incentive mechanism should motivate workers to participate in the tasks. In spatial crowdsourcing, different workers may have different interest in the task because of the variable spatial and temporal information of workers and tasks. Thus it has become a challenge how to design the incentive mechanism for spatial crowdsourcing.

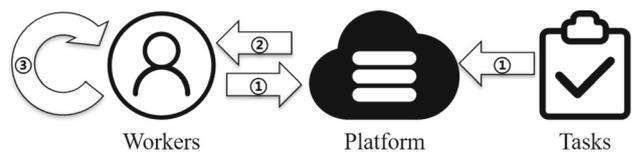
An incentive mechanism is assessed using two types of metrics, i.e., algorithm metrics and mechanism metrics. As shown in Table 8, most efforts focus on the algorithm metrics (especially the time complexity) of their mechanisms. Although many incentive mechanisms are computationally efficient (i.e., able to terminate in polynomial time), the spatiotemporal dynamics may raise a real-time requirement for practical incentive mechanism design. Mechanism metrics are emphasized more in auction models than in posted price models. This is because the auction-based model considers the participatory of the workers before pricing for workers and, as a result, requires the mechanism metrics to guarantee the robustness.

Besides, the formulation of the incentive models notably varies even for the applications (e.g., for taxi dispatching [45, 60, 210]). Hence it seems necessary to come up with a unified formulation such that the proposed incentive mechanisms can be fairly compared in terms of effectiveness, efficiency, and flexibility. Furthermore, many existing works focus on maximizing the revenue in short term, and it is still open how to design an incentive mechanism for the long-term revenue.

Finally, it is worth mentioning that there is another successful incentive besides the aforementioned ones: *volunteered-based incentive*. In practice, when the scale of the whole task is large (e.g., editing the whole map of world), it usually requires a large number of workers which often leads to a extremely high payment. Thus, a practical and efficient way to complete such tasks is to get help from volunteer workers [85]. For example, one of the biggest volunteer-based community in spatial crowdsourcing is the Humanitarian OpenStreetMap Team (HOT) [23]. Since its foundation in 2010, HOT has already had 170,252 registered volunteers and together completed 1,933,608 tasks related to environmental and societal issues (e.g., disaster response and risk reduction). The motivations of these volunteers are either contributing to the greater good (e.g., users in HOT) or gaining something by taking part (e.g., drivers in Waze [19]). However, rather than the algorithmic/theoretic aspects of incentive mechanisms, existing works on volunteer-based incentives usually focus on the supporting tool designs to attract volunteers [85, 137], which is not the major concern in this survey. We refer readers to [85, 136, 137, 152] on important issues in supporting tool designs for volunteer-based incentive mechanisms.

## 6 Privacy protection

As in traditional Web-based crowdsourcing, privacy is an important concern in spatial crowdsourcing. One particular interest in spatial crowdsourcing is to protect the location information of tasks and workers (and certain intermediate results) so that spatiotemporal tasks can be released and performed without exposing the physical locations of tasks



**Fig. 7** Workflow of privacy protection for task assignment in spatial crowdsourcing. Step (1): transformation; Step (2): assignment; Step (3): refinement

and workers to malicious users. Overall, privacy protection research in spatial crowdsourcing is dedicated to design privacy-preserving frameworks and techniques compatible for the core issues in spatial crowdsourcing (e.g., task assignment [171]).

### 6.1 Generic framework

Most studies on privacy protection in spatial crowdsourcing focus on privacy-preserving task assignment. A generic privacy-preserving framework for task assignment in spatial crowdsourcing consists of three steps. Figure 7 shows its workflow.

- (1) **Transformation** The locations of workers and (or) tasks are transformed by some techniques.
- (2) **Assignment** The spatial crowdsourcing platform performs task assignment based on the transformed locations of workers and (or) tasks.
- (3) **Refinement** Workers confirm or refine the task assignment results based on their true locations.

Depending on the location transformation techniques and the assumptions on trusted parties, the step of refinement may be omitted. Furthermore, some privacy protection schemes may involve auxiliary trusted servers. In the context of spatial crowdsourcing, the spatial crowdsourcing platform (the platform for short) is usually assumed to be untrusted.

Below we review representative studies that exploit three categories of transformation techniques: *spatiotemporal cloaking*, *differential privacy* and *encryption*.

### 6.2 Spatiotemporal cloaking-based transformation

Spatiotemporal cloaking protects location privacy by hiding the locations inside a cloaked region.

In [216], the locations of workers are first submitted to an extra trusted server. Then, the trusted server constructs a cloaked region around the worker's actual location for each worker based on locality-sensitive hashing (LSH) [76], where both  $K$ -anonymity [191] and locality are preserved. The untrusted spatial crowdsourcing platform can only access the above transformed spatial cloak of each worker. Then

an algorithm is devised for searching the  $k$ -nearest tasks of a worker with the help of the refinement by the trusted server, based on which task assignment can be performed.

In [130,131], the authors assume that the workers trust each other but do not trust the spatial crowdsourcing platform. Each worker calculates his/her Voronoi cell in a distributed manner and forms the spatial cloak. Then a voting mechanism is designed through which a set of representative participants are selected whose cloaked regions should be sent out to the spatial crowdsourcing platform for querying the nearest tasks, during with  $K$ -anonymity is preserved. These query results will later be shared with the rest of the workers. As a result, all the tasks are assigned to the nearest workers.

In [170], instead of a spatiotemporal point, each worker submits a cloaked area including a spatiotemporal region  $a$  and the probability density function  $f$  of the worker at each point in  $a$ . Based on the cloaked locations of workers and exact locations of tasks, the spatial crowdsourcing platform performs uncertain task pre-assignment via the expected distances between the tasks and workers. The pre-assignment results are sent to the service of the workers and refined according to the exact locations of the workers. Based on the methods in [170], [172] proposes a demo of a location-based mobile Q&A application.

### 6.3 Differential privacy-based transformation

Differential privacy [86] is a general-purpose approach for privacy protection and has emerged as the de facto standard for private data release. It releases data of a group such that what can be learned from the released data does not substantially differ regardless of the inclusion of a given individual's data [87]. Below are some works with different implementations of differential privacy for task assignment in spatial crowdsourcing.

In [196], the locations of workers are first submitted to a trusted server and then transformed via *Private Spatiotemporal Decomposition (PSD)* proposed in [71]. A PSD is a spatiotemporal index transformed according to differential privacy, where each index node is obtained by releasing a noisy count of the data points enclosed by that node's extent. The spatial crowdsourcing platform then performs task assignment as follows. For each task  $t$  the platform first queries the PSD released by the trusted server for a region where there are workers near  $t$  with a high probability. Then the platform geocasts the information of  $t$  to the workers in this region. The workers who are willing to perform  $t$  send a consent message back to the platform. The method can be also extended to dynamic data [200].

In [124,125], the authors investigate privacy protection in crowdsourced dynamic spectrum sensing. The locations of workers are not transformed directly. Instead, the bids

provided by all the workers are transformed, which represent the worker's cost for spectrum sensing and are closely tied to the worker's current locations. The transformation to protect differential privacy is based on the exponential mechanism [160]. Then task assignment is modeled as a reserve auction problem and privacy-preserving methods for worker selection are proposed based on the transformation.

In [217], the authors study privacy protection in crowdsourced urban sensing. To transform the locations of workers, the spatial crowdsourcing platform first provides an obfuscation matrix and a data adjustment function designed via differential privacy. The location information is transformed by the obfuscation matrix which encodes the probabilities of obfuscating any one region to another. The corresponding data are transformed by the data adjustment function. Note that the platform cannot obtain the original data although it provides the obfuscation matrix and data adjustment function. After the transformed locations and sensory data are uploaded to the spatial crowdsourcing platform, the platform can infer the distribution of the data in the sensed regions from the transformed data.

In [201], the authors propose a privacy-preserving one-sided online task assignment scheme where tasks appear dynamically. The locations of workers are transformed by Geo-indistinguishability (Geo-I) [35], which is a notion of location privacy based on differential privacy, and are submitted to the spatial crowdsourcing platform in advance. Once a task appears, the transformed location is submitted to the spatial crowdsourcing platform. Subsequently, the spatial crowdsourcing platform identifies a set of candidate workers who are most capable to perform the task and sends the information of these workers to the requester of the task. Finally, the requester of the task connects the workers one by one bypassing the spatial crowdsourcing platform and share the exact location to refine whom can complete the task. The authors test their privacy-preserving schemes on a taxi datasets called T-Drive [231]. Their experimental results show the proposed techniques, algorithms, and heuristics achieve high effectiveness and low disclosure of the location information.

### 6.4 Encryption-based transformation

This line of research applies specific encryption techniques on the locations of tasks and workers such that particular calculation can still be performed on the encrypted data, e.g., calculating the distance between two encrypted positions. Thus tasks can be assigned based on the calculation results.

In [150], the locations of tasks and workers are encrypted via a Paillier cryptosystem [169]. According to Paillier cryptosystem's characteristics, the exact distance between two encrypted positions can be calculated without releasing the plain data. Hence the distances between the tasks and work-

**Table 9** Comparison of privacy protection techniques in spatial crowdsourcing

Transformation techniques	Reference	Protected components	Arrival scenario	Assignment goal	Without extra server	Computation overhead
Spatiotemporal cloaking	[216]	Workers	Static	Nearest	✗	Low
	[130,131]	Workers	Static	Nearest	✓	low
	[170]	Workers	Static	Minimizing total cost	✓	Low
Differential privacy	[196]	Workers	Static	Maximizing total number	✗	Low
	[200]	Workers	Dynamic	Maximizing total number	✗	Low
	[124,125]	Workers	Static	Minimizing total cost	✓	Low
	[217]	Workers	Static	Quality of sensed data	✓	Low
Encryption	[201]	Tasks and workers	Dynamic	Maximizing total number	✓	Low
	[150]	Tasks and workers	Static	Maximizing total number and minimizing total cost	✗	High
	[149]	Tasks and workers	Static	Nearest	✗	High
	[148]	Tasks and workers	Static	Nearest	✗	High
	[233]	Workers	Static	Maximizing total utility	✗	High

ers can be obtained during task assignment. However, the adoption of the Paillier cryptosystem significantly increases the computation overhead. Thus the authors in [150] extend KD-tree to SKD-tree in order to prune unnecessary Paillier cryptosystem query or computation.

In [149], the authors take the velocity of workers into consideration and propose a Paillier cryptosystem [169] and ElGamal cryptosystem [95]-based privacy protection method. Specifically, the locations of tasks and workers are encrypted and the distance is computed in a privacy-preserving way. The velocity of workers is also encrypted and the travel time for a worker to move to the position of a task can also be computed with privacy preservation. Thus each task can be assignment to the worker with the minimum travel time.

In [148], researchers encrypt the locations of tasks and workers and calculate the distance secretly via the Paillier cryptosystem [169]. Distance comparison is performed via the Yao's protocol [146,229], and each task is assigned to the nearest worker. To improve the efficiency, Geohash [7] is adopted to find the nearest workers approximately.

In [175,233], the information of tasks including their positions are posed in the spatial crowdsourcing platform in advance and thus are not protected. The workers can browse the information and submit their travel costs for different tasks instead of their exact positions to the platform. The travel cost is encrypted into perturbed data via Bitwise XOR homomorphic cipher system [241]. After receiving all the perturbed data from the workers, the platform assigns tasks through a reverse auction-based algorithm.

## 6.5 Discussions

Table 9 shows representative studies on privacy protection using different transformation techniques. We summarize their main characteristics below.

- **Protected components** Spatial cloaking-based and most differential privacy-based methods only protect workers. The reasons are twofold. (i) The location privacy of workers is more sensitive than that of tasks. (ii) It is non-trivial to extend spatial cloaking and differential privacy-based transformation techniques to the cases where two components of data need to be protected.
- **Applicable task assignment categories** Most existing privacy-preserving task assignment schemes only apply to static arrival scenarios. Protecting dynamic location information is more difficult since many transformation techniques require all data to be protected to be known in advance. Since most studies integrate privacy protection into the task assignment framework, the applicable assignment goals are constrained by the privacy protection methods. For example, in [130,131,148,149,216] the goal is simply to assign the tasks to the nearest workers. For other more complex goals, the assignment methods are closely coupled with the privacy protection methods and most of them are heuristic.
- **Overhead** It adds extra overhead to protect privacy in task assignment. Encryption-based methods are often more computation-intensive than spatial cloaking and differential privacy-based methods, due to the high cost

of encryption/decryption and the need for extra trusted servers for key distribution.

- **Trade-off** The impact of enforcing privacy protection on the locations of workers and/or tasks depends on the specific privacy-preserving technique. If the transformation is performed through spatiotemporal cloaking or differential privacy, the effect is mainly on the number of assigned pairs of tasks and workers, as some pairs which satisfies the range constraint may not satisfy it anymore after their locations are transformed. However, the quality of each assigned pair is not affected due to the refinement step. If the transformation is performed through encryption, the quality is not impacted as all the calculation is exact. However, the low efficiency of encryption is its main drawback.

In summary, a practical privacy-preserving task assignment scheme should protect at least the locations of workers. Privacy protection brings extra overhead and constraints to task assignment in spatial crowdsourcing.

In addition to task assignment, privacy protection is also necessary in other core issues in spatial crowdsourcing. For example, privacy protection is combined with quality control in [217], where an inference algorithm is designed to improve the inference accuracy for the data transformed according to differential privacy. In [123], privacy protection is combined with incentive mechanism. Specifically, a reverse auction-based incentive mechanism is designed when considering the privacy requirement of different workers. Then, the data collected via spatial crowdsourcing is published after transformation through differential privacy.

## 7 Applications

Spatial crowdsourcing is closely tied to the physical world and there have been various real-world applications. This section summarizes typical spatial crowdsourcing applications into two categories: *sharing economy-based urban services* and *crowdsourced spatiotemporal data collection*.

### 7.1 Sharing economy-based urban services

Sharing economy-based urban services refer to applications such as delivery and onsite services crowdsourced to freelancers. In the context of spatial crowdsourcing, a task in these applications is usually served by a single worker and thus often involves no explicit quality control (i.e., result aggregation). Popular applications include on-demand taxi dispatching, ride sharing, food delivery and onsite microservices.

**On-demand taxi dispatching** It is one of the earliest successful spatial crowdsourcing applications. Passengers

appear dynamically and submit taxi requests to platforms such as Uber [17] and Didi Chuxing [4]. The platform assigns taxis to passengers in real time to pick up passengers. Hence, in terms of task assignment, on-demand taxi dispatching can be modeled as a dynamic matching problem with diverse objectives such as maximizing the total payoff or minimizing the average latency of passengers.

**Ride sharing** This application is an emerging extension of on-demand taxi dispatching service often provided by the same companies, e.g., Uber [17] and Didi Chuxing [4]. The key issue of ride sharing is to schedule a route, which consists of a sequence of pickup locations and delivery locations for each passenger to minimize the total travel cost of the drivers (i.e., workers) [158] or the average latency of the passengers (i.e., requesters) [227]. In terms of task assignment, ride sharing is often modeled as a dynamic planning problem [40,119,158,211].

**Food delivery** Food delivery services such as Grubhub [10] and Meituan [26] are similar to ride sharing in terms of task assignment. Customers dynamically submit food delivery requests to the platform. The platform then determines the price of the delivery requests for the requesters and the schedules of the delivery requests for the couriers. Similarly, food delivery services are often formulated as dynamic planning [154].

**Onsite microservices** Onsite microservices are another successful adoption of spatial crowdsourcing. Platforms such as TaskRabbit [15] and Gigwalk [8] connect various domestic services, e.g., house cleaning, with freelancers. Similar to on-demand taxi dispatching, task assignment in onsite microservices can be considered as a dynamic matching problem.

**Discussions** Sharing economy for urban services often deal with highly dynamic data at urban scale. To provide better quality of services, more efficient, and effective task assignment algorithms are needed. Sharing economy for urban services usually apply the incentive mechanisms based on supply-and-demand-aware models and provide certain degree of privacy protection. Nevertheless, there is a growing trend to introduce additional incentive mechanisms into these applications to consistently attract more users.

### 7.2 Crowdsourced spatiotemporal data collection

Crowdsourced spatiotemporal data collection refers to applications that crowdsource collection of various spatiotemporal information to citizens. In the context of spatial crowdsourcing, a task in these applications is usually performed by multiple workers and involves certain spatiotemporal data processing. Tasks in this category vary in real-time requirement and degree of spatiotemporal data processing, but all involve quality control to aggregate highly qualified results.

**Table 10** Categories of core issues in typical spatial crowdsourcing applications

Category	Reference	Task assignment	Quality control	Incentive mechanism	Privacy protection <sup>a</sup>
<i>Sharing economy-based urban services</i>					
Taxi dispatching	[4,17]	Dynamic matching	✗	Supply-and-demand-aware	✓
Ride sharing	[158,211]	Dynamic planning	✗	Supply-and-demand-aware	✓
Food delivery	[10,26]	Dynamic planning	✗	Supply-and-demand-aware	✓
Onsite microservices	[8,15,164]	Dynamic matching	✗	Supply-and-demand-aware	✓
<i>Spatiotemporal data collection</i>					
Event detection and labeling					
POI labeling	[117]	Static matching	Expectation maximization	✗	✗
Pollution detection	[109,188]	Static matching	Aggregated diversity	✗	✓
Event detection	[30,168]	Dynamic matching	Bayesian estimation	✗	✗
Map application					
Map generation	[62,107]	Dynamic matching	Aggregated diversity	Quality-aware	✓
Speed estimation	[116,155]	Static matching	Bayesian estimation	✗	✗
Road navigation	[19,89]	Static planning	Aggregated diversity	✗	✗
Congestion alert	[36]	Dynamic matching	Expectation maximization	✗	✗
Path selection	[189,234]	Static matching	Expectation maximization	✗	✗

<sup>a</sup>Some applications claim that privacy protection is considered but the detailed techniques are not specified

**Crowdsourced event detection and labeling** It is natural to crowdsource detecting and labeling of urban spots or events to citizens. For instance, residents can contribute to POI labeling in the neighborhood [117]. They can also report noise pollution [188], air pollution [109] and weather conditions [199] in the vicinity. Since such data are normally provided by unprofessional workers using noisy sensors, it is crucial to aggregate sensory data from workers to control the quality of the detection or labeling tasks. Truth inference is commonly used for quality control in crowdsourced event detection and labeling [162,168].

**Crowdsourced map applications** Spatial crowdsourcing can also be applied in more complex spatiotemporal data collection and processing such as map generation, real-time traffic speed estimation, and road navigation. For example, OpenStreetMap [107] is already the world's largest crowdsourced mapping project that creates a free and collaboratively editable map of the world. Real-time traffic speed in a map can be inferred by crowdsourcing speed estimation of a portion of seed roads and jointly considering historical speed information [116,155]. Crowdsourced road navigation is viable by collecting real-time traffic information, e.g., using Waze [19] and constructing a landmark scoring model for route recommendation [89]. Some other functions in map applications, such as alerting traffic congestion [36], or answering path selection queries [189,234] could also be crowdsourced by consulting nearby drivers and picking out desirable answers. Quality control in these applications is dedicated and sometimes is coupled with the underlying spatiotemporal data processing process.

**Discussions** Table 10 summarizes the aforementioned applications in spatial crowdsourcing. Compared with the sharing economy-based urban services, task assignment in crowdsourced spatiotemporal data collection depends on the specific data to collect and varies in the models. It can be formulated as static matching (e.g., POI labeling [117]), static planning (e.g., road navigation [89]), or dynamic matching (e.g., map generation [107]). It is important for crowdsourced spatiotemporal data collection applications to attract the highly qualified workers. Hence the incentive mechanisms based on the quality-aware models are used to motivate workers. While some pioneer studies have proposed privacy protection schemes for certain applications in this category (e.g., pollution detection [162] and map generation [62]), it is unclear whether privacy protection methods suited for other applications have been designed.

## 8 Open problems

In this section, we discuss some important open problems in spatial crowdsourcing.

**More effective task assignment algorithms** Task assignment is central to spatial crowdsourcing, yet its effectiveness is still not satisfactory for many real-world applications. Particularly, emerging applications such as on-demand taxi dispatching and ride sharing require highly effective dynamic matching and planning algorithms. Yet the competitive ratios of the state-of-the-art algorithms for dynamic matching are often no higher than 0.5 unless under some strong assump-

tions (e.g., one-sided [121], arrival rate [120]). It also seems hard to propose competitive solutions to dynamic task planning under extreme cases. In particular, the worst cases to prove the hardness result are usually impractical, e.g., with the extremely short deadline [144, 192]. Thus, existing studies (e.g., [119, 158, 192]) usually propose heuristics without any theoretical guarantee. One opportunity to improve the effectiveness of dynamic task assignment is that practical applications may not strictly require instant assignments. Therefore, it may be feasible to wait for a reasonably short period and make global assignments on a batch basis. However, it remains open how to theoretically select the best single batch or adapt the batch size in real time to notably improve the effectiveness of task assignment algorithms.

**Indices for spatial crowdsourced data** Efficient spatial crowdsourcing requires not only efficient algorithms but also efficient data structures, e.g., indices. Indices for spatial crowdsourced data need to be optimized for spatial queries and frequent updates. Some indices (e.g., grid, R-tree [106], quadtree [178] and  $k$ -d tree [49]) are proposed for spatial queries. Others are proposed to handle the dynamics of spatiotemporal data, such as 3D R-tree [193], HR-tree [165] and TPR-tree [177]. Recently, Jonathan et al. [126] exploit a pyramid multi-resolution index to speed up the retrieval of workers in a given area. However, dedicated spatiotemporal indices are overlooked in existing spatial crowdsourcing algorithms. It is largely unexplored how to select or design suitable spatiotemporal indices and co-optimize the end-to-end efficiency of spatial crowdsourcing algorithms.

**Benchmarks for spatial crowdsourcing** Standardized benchmarks are important for the continuous development of spatial crowdsourcing research. There have been many benchmarks for classical spatial data management. For example, DIMACS Implementation Challenge provides a set of benchmark instances for various shortest path problems. However, there is a lack of similar benchmarks for spatial crowdsourcing. Although there are a few synthetic data generators for spatial crowdsourcing [198], the lack of public real-world datasets still presents a challenge to the development of spatial crowdsourcing. The reasons of such a quandary are twofold. First, the owners of real data are usually commercial platforms that are not willing to share their data. Secondly, although there are open-source spatial crowdsourcing platforms such as gMission [63] and MediaQ [25], they cannot collect large amounts of data due to their limited scales.

## 9 Conclusion

In this paper, we surveyed the state-of-the-art research on spatial crowdsourcing, with comprehensive comparisons between spatial crowdsourcing and general-purposed crowd-

sourcing in terms of challenges and techniques. We summarized existing literature on spatial crowdsourcing algorithms into four categories: task assignment, quality control, incentive mechanism design, and privacy protection. Particularly, for task assignment, we reviewed matching and planning models in static and dynamic scenarios; for quality control, we discussed quality models of tasks/workers and result aggregation techniques; for incentive mechanism design, we presented posted price models and auction-based models; for privacy protection, we offered a general privacy protection framework and compared three types of data transformation techniques. In addition, we studied emerging representative spatial crowdsourcing applications and explained how they are enabled by these techniques. Finally, we identified some open problems for future research in this active research area. We envision this survey as a timely reference and guideline for researchers and practitioners in spatial crowdsourcing.

**Acknowledgements** We are grateful to anonymous reviewers for their constructive comments. Yongxin Tong's work is partially supported by the National Science Foundation of China (NSFC) under Grant Nos. 61822201, U1811463 and 71531001, Science and Technology Major Project of Beijing under Grant Nos. Z171100005117001 and Didi Gaia Collaborative Research Funds for Young Scholars. Yuxiang Zeng and Lei Chen's works are partially supported by the Hong Kong RGC CRF C6030-18G Project, the National Science Foundation of China (NSFC) under Grant No. 61729201, Science and Technology Planning Project of Guangdong Province, China, No. 2015B010110006, Hong Kong ITC ITF Grants ITS/212/16FP and ITS/470/18FX, Didi-HKUST joint research lab project, Microsoft Research Asia Collaborative Research Grant and Wechat Research Grant. Cyrus Shahabi's work has been funded in part by NSF Grants IIS1320149 and CNS-1461963, the USC Integrated Media Systems Center (IMSC), and unrestricted cash gifts from Google and Oracle. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of any of the sponsors such as the National Science Foundation.

## References

1. Datatang Taxi Dataset (2016). <http://www.datatang.com/data/45888>. Accessed 23 June 2016
2. Amazon Mechanical Turk (2018). <https://www.mturk.com/>. Accessed 26 Dec 2018
3. Cainiao (2018). <https://www.cainiao.com/>. Accessed 26 Dec 2018
4. Didi Chuxing (2018). <https://www.didiglobal.com/>. Accessed 26 Dec 2018
5. Facebook Editor (2018). <https://www.facebook.com/editor>. Accessed 26 Dec 2018
6. FedEx (2018). <https://www.fedex.com/>. Accessed 26 Dec 2018
7. Geohash (2018). <https://en.wikipedia.org/wiki/Geohash>. Accessed 26 Dec 2018
8. Gigwalk (2018). <http://www.gigwalk.com>. Accessed 26 Dec 2018
9. gMission Dataset Generator (2018). <https://github.com/gmission/SCDataGenerator>. Accessed 26 Dec 2018
10. GrubHub (2018). <https://www.grubhub.com/>. Accessed 26 Dec 2018
11. InterestingSport (2018). <http://www.quyundong.com/>. Accessed 26 Dec 2018

12. Nanguache (2018). <http://www.nanguache.com/>. Accessed 26 Dec 2018
13. OpenStreetMap (2018). <https://www.openstreetmap.org/>. Accessed 26 Dec 2018
14. Pokémon Go (2018). <https://www.pokemongo.com/>. Accessed 26 Dec 2018
15. TaskRabbit (2018). <http://www.taskrabbit.com>. Accessed 26 Dec 2018
16. TLC Trip Record Data (2018). [http://www.nyc.gov/html/tlc/html/about/trip\\_record\\_data.shtml](http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml). Accessed 23 June 2016. Accessed 26 Dec 2018
17. Uber (2018). <https://www.uber.com/>. Accessed 26 Dec 2018
18. UPS (2018). <https://www.ups.com/>. Accessed 26 Dec 2018
19. Waze (2018). <http://www.waze.com/>. Accessed 26 Dec 2018
20. CPLEX (2019). <https://www.ibm.com/analytics/cplex-optimizer>. Accessed 26 May 2019
21. Didi Chuxing Corporate Citizenship Report (2019). <https://www.didiglobal.com/about-didi/responsibility>. Accessed 26 May 2019
22. GAIA Open Dataset (2019). <https://outreach.didichuxing.com/research/opensource>. Accessed 26 May 2019
23. Humanitarian OpenStreetMap Team (2019). <https://www.hotosm.org/>. Accessed 26 May 2019
24. keepright (2019). <https://www.keepright.at/>. Accessed 26 May 2019
25. MediaQ (2019). <http://mediaq.usc.edu/>. Accessed 26 May 2019
26. Meituan (2019). <https://www.meituan.com/>. Accessed 26 May 2019
27. Seamless (2019). <https://www.seamless.com>. Accessed 26 May 2019
28. Upwork (2019). <https://www.upwork.com/>. Accessed 26 May 2019
29. Wikimapia (2019). <https://www.wikimapia.org/>. Accessed 26 May 2019
30. Agapie, E., Teevan, J., Monroy-Hernández, A.: Crowdsourcing in the field: A case study using local crowds for event reporting. In: Proceedings of the 3rd AAAI Conference on Human Computation and Crowdsourcing, pp. 2–11 (2015)
31. Aggarwal, G., Goel, G., Karande, C., Mehta, A.: Online vertex-weighted bipartite matching and single-bid budgeted allocations. In: Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1253–1264 (2011)
32. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms and Applications. Prentice Hall, Upper Saddle River (1993)
33. Alfarrarjeh, A., Emrich, T., Shahabi, C.: Scalable spatial crowdsourcing: A study of distributed algorithms. In: 16th IEEE International Conference on Mobile Data Management, pp. 134–144 (2015)
34. Amsterdamer, Y., Milo, T.: Foundations of crowd data sourcing. SIGMOD Record **43**(4), 5–14 (2014)
35. Andrés, M.E., Bordenabe, N.E., Chatzikokolakis, K., Palamidessi, C.: Geo-indistinguishability: differential privacy for location-based systems. In: 2013 ACM SIGSAC Conference on Computer and Communications Security, pp. 901–914 (2013)
36. Artikis, A., Weidlich, M., Schnitzler, F., Boutsis, I., Liebig, T., Piatkowski, N., Bockermann, C., Morik, K., Kalogeraki, V., Marecek, J., Gal, A., Mannor, S., Gunopulos, D., Kinane, D.: Heterogeneous stream processing and crowdsourcing for urban traffic management. In: Proceedings of the 17th International Conference on Extending Database Technology, pp. 712–723 (2014)
37. Asghari, M., Shahabi, C.: An on-line truthful and individually rational pricing mechanism for ride-sharing. In: Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 7:1–7:10 (2017)
38. Asghari, M., Shahabi, C.: On on-line task assignment in spatial crowdsourcing. In: 2017 IEEE International Conference on Big Data, pp. 395–404 (2017)
39. Asghari, M., Shahabi, C.: Adapt-pricing: a dynamic and predictive technique for pricing to maximize revenue in ridesharing platforms. In: Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 189–198 (2018)
40. Asghari, M., Deng, D., Shahabi, C., Demiryurek, U., Li, Y.: Price-aware real-time ride-sharing at scale: an auction-based approach. In: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 3:1–3:10 (2016)
41. Ashlagi, I., Azar, Y., Charikar, M., Chiplunkar, A., Geri, O., Kaplan, H., Makhijani, R.M., Wang, Y., Wattenhofer, R.: Min-cost bipartite perfect matching with delays. In: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017), pp. 1:1–1:20 (2017)
42. Azar, Y., Fanani, A.J.: Deterministic min-cost matching with delays. In: 16th International Workshop on Approximation and Online Algorithms, pp. 21–35 (2018)
43. Azar, Y., Chiplunkar, A., Kaplan, H.: Polylogarithmic bounds on the competitiveness of min-cost perfect matching with delays. In: Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1051–1061 (2017)
44. Banerjee, S., Johari, R., Riquelme, C.: Pricing in ride-sharing platforms: A queueing-theoretic approach. In: Proceedings of the 16th ACM Conference on Economics and Computation, p. 639 (2015)
45. Banerjee, S., Freund, D., Lykouris, T.: Pricing and optimization in shared vehicle systems: An approximation framework. In: Proceedings of the 2017 ACM Conference on Economics and Computation, p. 517 (2017)
46. Bansal, N., Buchbinder, N., Gupta, A., Naor, J.: A randomized  $o(\log^2 k)$ -competitive algorithm for metric bipartite matching. Algorithmica **68**(2), 390–403 (2014)
47. Bar-Yehuda, R., Bendel, K., Freund, A., Rawitz, D.: Local ratio: a unified framework for approximation algorithms. ACM Comput. Surv. **36**(4), 422–463 (2004)
48. Bei, X., Zhang, S.: Algorithms for trip-vehicle assignment in ride-sharing. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, pp. 3–9 (2018)
49. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Commun. ACM **18**(9), 509–517 (1975)
50. Birnbaum, B.E., Mathieu, C.: On-line bipartite matching made simple. SIGACT News **39**(1), 80–87 (2008)
51. Brubach, B., Sankararaman, K.A., Srinivasan, A., Xu, P.: New algorithms, better bounds, and a novel model for online stochastic matching. In: 24th Annual European Symposium on Algorithms, pp. 24:1–24:16 (2016)
52. Burkard, R.E., Dell’Amico, M., Martello, S.: Assignment problems. Springer, Berlin (2009)
53. Cao, C.C., She, J., Tong, Y., Chen, L.: Whom to ask? Jury selection for decision making tasks on micro-blog services. PVLDB **5**(11), 1495–1506 (2012)
54. Castillo, J., Knoepfle, D., Weyl, G.: Surge pricing solves the wild goose chase. In: Proceedings of the 2017 ACM Conference on Economics and Computation, pp. 241–242 (2017)
55. Chen, C., Cheng, S., Misra, A., Lau, H.C.: Multi-agent task assignment for mobile crowdsourcing under trajectory uncertainties. In: Proceedings of the 14th International Conference on Autonomous Agents and MultiAgent Systems, pp. 1715–1716 (2015)
56. Chen, J., Zipf, A.: Deepvgti: Deep learning with volunteered geographic information. In: Proceedings of the 26th International Conference on World Wide Web Companion, pp. 771–772 (2017)

57. Chen, L., Shahabi, C.: Spatial crowdsourcing: challenges and opportunities. *IEEE Data Eng. Bull.* **39**(4), 14–25 (2016)
58. Chen, L., Lee, D., Zhang, M.: Crowdsourcing in information and knowledge management. In: *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management* (2014)
59. Chen, L., Lee, D., Milo, T.: Data-driven crowdsourcing: Management, mining, and applications. In: *31st IEEE International Conference on Data Engineering*, pp. 1527–1529 (2015)
60. Chen, M., Shen, W., Tang, P., Zuo, S.: Optimal vehicle dispatching for ride-sharing platforms via dynamic pricing. In: *Companion of The Web Conference*, pp. 51–52 (2018)
61. Chen, M.K.: Dynamic pricing in a labor market: Surge pricing and flexible work on the uber platform. In: *Proceedings of the 2016 ACM Conference on Economics and Computation*, p. 455 (2016)
62. Chen, X., Wu, X., Li, X., Ji, X., He, Y., Liu, Y.: Privacy-aware high-quality map generation with participatory sensing. *IEEE Trans. Mob. Comput.* **15**(3), 719–732 (2016)
63. Chen, Z., Fu, R., Zhao, Z., Liu, Z., Xia, L., Chen, L., Cheng, P., Cao, C.C., Tong, Y., Zhang, C.J.: gMission: a general spatial crowdsourcing platform. *PVLDB* **7**(13), 1629–1632 (2014)
64. Chen, Z., Cheng, P., Zeng, Y., Chen, L.: Minimizing maximum delay of task assignment in spatial crowdsourcing. In: *35th IEEE International Conference on Data Engineering*, pp. 1454–1465 (2019)
65. Cheng, P., Lian, X., Chen, Z., Fu, R., Chen, L., Han, J., Zhao, J.: Reliable diversity-based spatial crowdsourcing by moving workers. *PVLDB* **8**(10), 1022–1033 (2015)
66. Cheng, P., Lian, X., Chen, L., Han, J., Zhao, J.: Task assignment on multi-skill oriented spatial crowdsourcing. *IEEE Trans. Knowl. Data Eng.* **28**(8), 2201–2215 (2016)
67. Cheng, P., Jian, X., Chen, L.: An experimental evaluation of task assignment in spatial crowdsourcing. *PVLDB* **11**(11), 1428–1440 (2018)
68. Cheng, S., Nguyen, D.T., Lau, H.C.: Mechanisms for arranging ride sharing and fare splitting for last-mile travel demands. In: *Proceedings of the 13th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1505–1506 (2014)
69. Chittilappilly, A.I., Chen, L., Amer-Yahia, S.: A survey of general-purpose crowdsourcing techniques. *IEEE Trans. Knowl. Data Eng.* **28**(9), 2246–2266 (2016)
70. Chuang, T., Deng, D., Hsu, C., Lemmens, R.: The one and many maps: participatory and temporal diversities in openstreetmap. In: *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information*, pp. 79–86 (2013)
71. Cormode, G., Procopiuc, C.M., Srivastava, D., Shen, E., Yu, T.: Differentially private spatial decompositions. In: *28th IEEE International Conference on Data Engineering*, pp. 20–31 (2012)
72. Corral, A., Manolopoulos, Y., Theodoridis, Y., Vassilakopoulos, M.: Closest pair queries in spatial databases. In: *Proceedings of the 2000 ACM International Conference on Management of Data*, pp. 189–200 (2000)
73. Costa, C.F., Nascimento, M.A.: In-route task selection in crowdsourcing. In: *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 524–527 (2018)
74. Cranshaw, J., Toch, E., Hong, J.I., Kittur, A., Sadeh, N.M.: Bridging the gap between physical location and online social networks. In: *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, pp. 119–128 (2010)
75. Das, A., Gollapudi, S., Kim, A., Panigrahi, D., Swamy, C.: Minimizing latency in online ride and delivery services. In: *Proceedings of the 27th International Conference on World Wide Web*, pp. 379–388 (2018)
76. Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.S.: Locality-sensitive hashing scheme based on p-stable distributions. In: *Proceedings of the 20th ACM Symposium on Computational Geometry*, pp. 253–262 (2004)
77. Demartini, G., Difallah, D.E., Cudré-Mauroux, P.: Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In: *Proceedings of the 21st International Conference on World Wide Web*, pp. 469–478 (2012)
78. Deng, D., Shahabi, C., Demiryurek, U.: Maximizing the number of worker’s self-selected tasks in spatial crowdsourcing. In: *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 314–323 (2013)
79. Deng, D., Shahabi, C., Zhu, L.: Task matching and scheduling for multiple workers in spatial crowdsourcing. In: *Proceedings of the 23rd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 21:1–21:10 (2015)
80. Deng, D., Shahabi, C., Demiryurek, U., Zhu, L.: Task selection in spatial crowdsourcing from worker’s perspective. *GeoInformatica* **20**(3), 529–568 (2016)
81. Derigs, U.: A shortest augmenting path method for solving minimal perfect matching problems. *Networks* **11**(4), 379–390 (1981)
82. Devanur, N.R., Jain, K., Kleinberg, R.D.: Randomized primal-dual analysis of RANKING for online bipartite matching. In: *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 101–107 (2013)
83. Dickerson, J.P., Sankararaman, K.A., Srinivasan, A., Xu, P.: Allocation problems in ride-sharing platforms: Online matching with offline reusable resources. In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pp. 1007–1014 (2018)
84. Dickerson, J.P., Sankararaman, K.A., Srinivasan, A., Xu, P.: Assigning tasks to workers based on historical data: Online task assignment with two-sided arrivals. In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 318–326 (2018)
85. Dittus, M., Quattrone, G., Capra, L.: Analysing volunteer engagement in humanitarian mapping: building contributor communities at large scale. In: *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pp. 108–118 (2016)
86. Dwork, C.: Differential privacy. In: *International Colloquium on Automata, Languages and Programming*, pp. 1–12 (2006)
87. Dwork, C.: Differential privacy: a survey of results. In: *5th International Conference on Theory and Applications of Models of Computation*, pp. 1–19 (2008)
88. Emek, Y., Kuten, S., Wattenhofer, R.: Online matching: haste makes waste! In: *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 333–344 (2016)
89. Fan, X., Liu, J., Wang, Z., Jiang, Y., Liu, X.: Crowdsourced road navigation: concept, design, and implementation. *IEEE Commun. Mag.* **55**(6), 126–128 (2017)
90. Fang, Z., Huang, L., Wierman, A.: Prices and subsidies in the sharing economy. In: *Proceedings of the 26th International Conference on World Wide Web*, pp. 53–62 (2017)
91. Feldman, J., Mehta, A., Mirrokni, V.S., Muthukrishnan, S.: Online stochastic matching: Beating 1-1/e. In: *50th Annual IEEE Symposium on Foundations of Computer Science*, pp. 117–126 (2009)
92. Ferguson, T.S., et al.: Who solved the secretary problem? *Stat. Sci.* **4**(3), 282–289 (1989)
93. Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. *Can. J. Math.* **8**(3), 399–404 (1956)
94. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. *Am. Math. Mon.* **69**(1), 9–15 (1962)
95. Gamal, T.E.: A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Theory* **31**(4), 469–472 (1985)

96. Gao, D., Tong, Y., She, J., Song, T., Chen, L., Xu, K.: Top-k team recommendation in spatial crowdsourcing. In: 17th International Conference on Web-Age Information Management, pp. 191–204 (2016)
97. Gao, D., Tong, Y., Ji, Y., Xu, K.: Team-oriented task planning in spatial crowdsourcing. In: Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint Conference on Web and Big Data, pp. 41–56 (2017)
98. Gao, D., Tong, Y., She, J., Song, T., Chen, L., Xu, K.: Top-k team recommendation and its variants in spatial crowdsourcing. *Data Sci. Eng.* **2**(2), 136–150 (2017)
99. Garcia-Molina, H., Joglekar, M., Marcus, A., Parameswaran, A.G., Verroios, V.: Challenges in data crowdsourcing. *IEEE Trans. Knowl. Data Eng.* **28**(4), 901–911 (2016)
100. Garcia-Ulloa, D.A., Xiong, L., Sunderam, V.S.: Truth discovery for spatiotemporal events from crowdsourced data. *PVLDB* **10**(11), 1562–1573 (2017)
101. Garey, M.R., Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco (1979)
102. Goel, G., Mehta, A.: Online budgeted matching in random input models with applications to adwords. In: Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 982–991 (2008)
103. Gu, L., Wang, K., Liu, X., Guo, S., Liu, B.: A reliable task assignment strategy for spatial crowdsourcing in big data environment. In: IEEE International Conference on Communications, pp. 1–6 (2017)
104. Guo, B., Liu, Y., Wang, L., Li, V.O.K., Lam, J.C.K., Yu, Z.: Task allocation in spatial crowdsourcing: Current state and future directions. *IEEE Internet Things J.* **5**(3), 1749–1764 (2018)
105. Guo, S., Parameswaran, A.G., Garcia-Molina, H.: So who won?: dynamic max discovery with the crowd. In: Proceedings of the 2012 ACM International Conference on Management of Data, pp. 385–396 (2012)
106. Guttman, A.: R-trees: A dynamic index structure for spatial searching. In: Proceedings of the 1984 ACM International Conference on Management of Data, pp. 47–57 (1984)
107. Haklay, M.M., Weber, P.: Openstreetmap: user-generated street maps. *IEEE Pervasive Comput.* **7**(4), 12–18 (2008)
108. Han, S., Xu, Z., Zeng, Y., Chen, L.: Fluid: A blockchain based framework for crowdsourcing. In: Proceedings of the 2019 ACM International Conference on Management of Data, pp. 1921–1924 (2019)
109. Hasenfratz, D., Saukh, O., Sturzenegger, S., Thiele, L.: Participatory air pollution monitoring using smartphones. *Mob. Sens.* **1**, 1–5 (2012)
110. Hashemi, P., Abbaspour, R.A.: Assessment of logical consistency in openstreetmap based on the spatial similarity concept. In: *OpenStreetMap in GIScience, Lecture Notes in Geoinformation and Cartography*, pp. 19–36. Springer, Berlin (2015)
111. ul Hassan, U., Curry, E.: A multi-armed bandit approach to online spatial task assignment. In: 2014 IEEE 11th International Conference on Ubiquitous Intelligence and Computing and 2014 IEEE 11th International Conference on Autonomic and Trusted Computing and 2014 IEEE 14th International Conference on Scalable Computing and Communications and Its Associated Workshops, pp. 212–219 (2014)
112. ul Hassan, U., Curry, E.: Efficient task assignment for spatial crowdsourcing: a combinatorial fractional optimization approach with semi-bandit learning. *Expert Syst. Appl.* **58**, 36–56 (2016)
113. He, S., Shin, D., Zhang, J., Chen, J.: Toward optimal allocation of location dependent tasks in crowdsensing. In: 2014 IEEE Conference on Computer Communications, pp. 745–753 (2014)
114. Heipke, C.: Crowdsourcing geospatial data. *ISPRS J. Photogramm. Remote Sens.* **65**(6), 550–557 (2010)
115. Ho, C., Jabbari, S., Vaughan, J.W.: Adaptive task assignment for crowdsourced classification. In: Proceedings of the 30th International Conference on Machine Learning, pp. 534–542 (2013)
116. Hu, H., Li, G., Bao, Z., Cui, Y., Feng, J.: Crowdsourcing-based real-time urban traffic speed estimation: from trends to speeds. In: 32nd IEEE International Conference on Data Engineering, pp. 883–894 (2016)
117. Hu, H., Zheng, Y., Bao, Z., Li, G., Feng, J., Cheng, R.: Crowd-sourced POI labelling: location-aware result inference and task assignment. In: 32nd IEEE International Conference on Data Engineering, pp. 61–72 (2016)
118. Huang, P., Zhu, W., Liao, K., Sellis, T., Yu, Z., Guo, L.: Efficient algorithms for flexible sweep coverage in crowdsensing. *IEEE Access* **6**, 50055–50065 (2018)
119. Huang, Y., Bastani, F., Jin, R., Wang, X.S.: Large scale real-time ridesharing with service guarantee on road networks. *PVLDB* **7**(14), 2017–2028 (2014)
120. Huang, Z., Kang, N., Tang, Z.G., Wu, X., Zhang, Y., Zhu, X.: How to match when all vertices arrive online. In: Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, pp. 17–29 (2018)
121. Jaillet, P., Lu, X.: Online stochastic matching: new algorithms with better bounds. *Math. Oper. Res.* **39**(3), 624–646 (2014)
122. Jiang, S., Chen, L., Mislove, A., Wilson, C.: On ridesharing competition and accessibility: Evidence from uber, lyft, and taxi. In: Proceedings of the 2018 International Conference on World Wide Web, pp. 863–872 (2018)
123. Jin, H., Su, L., Xiao, H., Nahrstedt, K.: Incentive mechanism for privacy-aware data aggregation in mobile crowd sensing systems. *IEEE/ACM Trans. Netw.* **26**(5), 2019–2032 (2018)
124. Jin, X., Zhang, Y.: Privacy-preserving crowdsourced spectrum sensing. In: Proceedings of the IEEE International Conference on Computer Communications, pp. 1–9 (2016)
125. Jin, X., Zhang, Y.: Privacy-preserving crowdsourced spectrum sensing. *IEEE/ACM Trans. Netw.* **26**(3), 1236–1249 (2018)
126. Jonathan, C., Mokbel, M.F.: Stella: geotagging images via crowdsourcing. In: Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 169–178 (2018)
127. Kalyanasundaram, B., Pruhs, K.: Online weighted matching. *J. Algorithms* **14**(3), 478–488 (1993)
128. Kalyanasundaram, B., Pruhs, K.: An optimal deterministic algorithm for online b-matching. *Theor. Comput. Sci.* **233**(1–2), 319–325 (2000)
129. Karp, R.M., Vazirani, U.V., Vazirani, V.V.: An optimal algorithm for on-line bipartite matching. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, pp. 352–358 (1990)
130. Kazemi, L., Shahabi, C.: A privacy-aware framework for participatory sensing. *SIGKDD Explor.* **13**(1), 43–51 (2011)
131. Kazemi, L., Shahabi, C.: Towards preserving privacy in participatory sensing. In: 9th Annual IEEE International Conference on Pervasive Computing and Communications, pp. 328–331 (2011)
132. Kazemi, L., Shahabi, C.: Geocrowd: enabling query answering with spatial crowdsourcing. In: Proceedings of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 189–198 (2012)
133. Kazemi, L., Shahabi, C., Chen, L.: Geotrucrowd: trustworthy query answering with spatial crowdsourcing. In: Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 304–313 (2013)
134. Kesselheim, T., Radke, K., Tönnis, A., Vöcking, B.: An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In: 21st Annual European Symposium on Algorithms, pp. 589–600 (2013)

135. Khuller, S., Mitchell, S.G., Vazirani, V.V.: On-line algorithms for weighted bipartite matching and stable marriages. *Theor. Comput. Sci.* **127**(2), 255–267 (1994)
136. Kim, S., Mankoff, J., Paulos, E.: Sensr: evaluating a flexible framework for authoring mobile data-collection tools for citizen science. In: *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, pp. 1453–1462 (2013)
137. Kim, S., Mankoff, J., Paulos, E.: Exploring barriers to the adoption of mobile technologies for volunteer data collection campaigns. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 3117–3126 (2015)
138. Kooti, F., Grbovic, M., Aiello, L.M., Djuric, N., Radosavljevic, V., Lerman, K.: Analyzing uber's ride-sharing economy. In: *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 574–582 (2017)
139. Korula, N., Pál, M.: Algorithms for secretary problems on graphs and hypergraphs. In: *36th International Colloquium on Automata, Languages, and Programming*, pp. 508–520 (2009)
140. Kuncheva, L.I., Whitaker, C.J., Shipp, C.A., Duin, R.P.W.: Limits on the majority vote accuracy in classifier fusion. *Pattern Anal. Appl.* **6**(1), 22–31 (2003)
141. Li, G., Wang, J., Zheng, Y., Franklin, M.J.: Crowdsourced data management: a survey. *IEEE Trans. Knowl. Data Eng.* **28**(9), 2296–2319 (2016)
142. Li, G., Zheng, Y., Fan, J., Wang, J., Cheng, R.: Crowdsourced data management: Overview and challenges. In: *Proceedings of the 2017 ACM International Conference on Management of Data*, pp. 1711–1716 (2017)
143. Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation. In: *Proceedings of the 19th International Conference on World Wide Web*, pp. 661–670 (2010)
144. Li, Y., Yiu, M.L., Xu, W.: Oriented online route recommendation for spatial crowdsourcing task workers. In: *International Symposium on Spatial and Temporal Databases*, pp. 137–156 (2015)
145. Li, Y., Fang, J., Zeng, Y., Maag, B., Tong, Y., Zhang, L.: Two-sided online bipartite matching in spatial data: experiments and analysis. *GeoInformatica* (2019). <https://doi.org/10.1007/s10707-019-00359-w>
146. Lindell, Y., Pinkas, B.: A proof of security of Yao's protocol for two-party computation. *J. Cryptol.* **22**(2), 161–188 (2009)
147. Littlestone, N., Warmuth, M.K.: The weighted majority algorithm. *Inf. Comput.* **108**(2), 212–261 (1994)
148. Liu, A., Li, Z., Liu, G., Zheng, K., Zhang, M., Li, Q., Zhang, X.: Privacy-preserving task assignment in spatial crowdsourcing. *J. Comput. Sci. Technol.* **32**(5), 905–918 (2017)
149. Liu, A., Wang, W., Shang, S., Li, Q., Zhang, X.: Efficient task assignment in spatial crowdsourcing with worker and task privacy protection. *GeoInformatica* **22**(2), 335–362 (2018)
150. Liu, B., Chen, L., Zhu, X., Zhang, Y., Zhang, C., Qiu, W.: Protecting location privacy in spatial crowdsourcing using encrypted data. In: *Proceedings of the 20th International Conference on Extending Database Technology*, pp. 478–481 (2017)
151. Liu, J., Ji, Y., Lv, W., Xu, K.: Budget-aware dynamic incentive mechanism in spatial crowdsourcing. *J. Comput. Sci. Technol.* **32**(5), 890–904 (2017)
152. Liu, S.B., Iacucci, A.A., Meier, P.: Ushahidi haiti and chile: next generation crisis mapping. *ACSM Bulletin* 246 (2010)
153. Liu, X., He, Q., Tian, Y., Lee, W., McPherson, J., Han, J.: Event-based social networks: linking the online and offline social worlds. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1032–1040 (2012)
154. Liu, Y., Guo, B., Du, H., Yu, Z., Zhang, D., Chen, C.: Foodnet: Optimized on demand take-out food delivery using spatial crowdsourcing. In: *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pp. 564–566 (2017)
155. Liu, Z., Chen, L., Tong, Y.: Realtime traffic speed estimation with sparse crowdsourced data. In: *34th IEEE International Conference on Data Engineering*, pp. 329–340 (2018)
156. Long, C., Wong, R.C., Yu, P.S., Jiang, M.: On optimal worst-case matching. In: *Proceedings of the 2013 ACM International Conference on Management of Data*, pp. 845–856 (2013)
157. Lu, A., Frazier, P.I., Kislev, O.: Surge pricing moves uber's driver-partners. In: *Proceedings of the 2018 ACM Conference on Economics and Computation*, p. 3 (2018)
158. Ma, S., Zheng, Y., Wolfson, O.: T-share: A large-scale dynamic taxi ridesharing service. In: *29th IEEE International Conference on Data Engineering*, pp. 410–421 (2013)
159. Manshadi, V.H., Gharan, S.O., Saberi, A.: Online stochastic matching: online actions based on offline statistics. In: *Proceedings of the 32nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1285–1294 (2011)
160. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: *48th Annual IEEE Symposium on Foundations of Computer Science*, pp. 94–103 (2007)
161. Meyerson, A., Nanavati, A., Poplawski, L.J.: Randomized online algorithms for minimum metric bipartite matching. In: *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 954–959 (2006)
162. Minerand, J., Lancerin, F., Balasubramaniam, S., Conti, M., Tarkoma, S.: You are airing too much: assessing the privacy of users in crowdsourcing environmental data. In: *2015 IEEE TrustCom/BigDataSE/ISPA*, pp. 523–530 (2015)
163. Mitsopoulou, E., Boutsis, I., Kalogeraki, V., Yu, J.Y.: A cost-aware incentive mechanism in mobile crowdsourcing systems. In: *2018 19th IEEE International Conference on Mobile Data Management*, pp. 239–244 (2018)
164. Musthag, M., Ganesan, D.: Labor dynamics in a mobile micro-task market. In: *2013 ACM SIGCHI Conference on Human Factors in Computing Systems*, pp. 641–650 (2013)
165. Nascimento, M.A., Silva, J.R.O.: Towards historical r-trees. In: *Proceedings of the 1998 ACM Symposium on Applied Computing*, pp. 235–240 (1998)
166. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions—i. *Math. Program.* **14**(1), 265–294 (1978)
167. Ouyang, W.R., Srivastava, M.B., Toniolo, A., Norman, T.J.: Truth discovery in crowdsourced detection of spatial events. In: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 461–470 (2014)
168. Ouyang, W.R., Srivastava, M.B., Toniolo, A., Norman, T.J.: Truth discovery in crowdsourced detection of spatial events. *IEEE Trans. Knowl. Data Eng.* **28**(4), 1047–1060 (2016)
169. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 223–238 (1999)
170. Pournajaf, L., Xiong, L., Sunderam, V.S., Goryczka, S.: Spatial task assignment for crowd sensing with cloaked locations. In: *IEEE 15th International Conference on Mobile Data Management*, pp. 73–82 (2014)
171. Pournajaf, L., Garcia-Ulloa, D.A., Xiong, L., Sunderam, V.S.: Participant privacy in mobile crowd sensing task management: A survey of methods and challenges. In: *Proceedings of the 2015 ACM International Conference on Management of Data*, vol. 44, no. 4, pp. 23–34 (2015)
172. Pournajaf, L., Xiong, L., Sunderam, V.S., Xu, X.: STAC: spatial task assignment for crowd sensing with cloaked participant locations. In: *Proceedings of the 23rd ACM SIGSPATIAL Inter-*

- national Conference on Advances in Geographic Information Systems, pp. 90:1–90:4 (2015)
173. Quattrone, G., Mashhadi, A., Capra, L.: Mind the map: the impact of culture and economic affluence on crowd-mapping behaviours. In: Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work and Social Computing, pp. 934–944 (2014)
  174. Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. *J. Mach. Learn. Res.* **11**, 1297–1322 (2010)
  175. Ren, X., Yu, C., Yu, W., Yang, S., Yang, X., McCann, J.A., Yu, P.S.: Lopub: high-dimensional crowdsourced data publication with local differential privacy. *IEEE Trans. Inf. Forensics Secur.* **13**(9), 2151–2166 (2018)
  176. Robbins, H.: Some aspects of the sequential design of experiments. *Bull. Am. Math. Soc.* **58**(5), 527–535 (1952)
  177. Saltenis, S., Jensen, C.S., Leutenegger, S.T., López, M.A.: Indexing the positions of continuously moving objects. In: Proceedings of the 2000 ACM International Conference on Management of Data, pp. 331–342 (2000)
  178. Samet, H.: The quadtree and related hierarchical data structures. *ACM Comput. Surv.* **16**(2), 187–260 (1984)
  179. Senaratne, H., Mobasheri, A., Ali, A.L., Capineri, C., Haklay, M.: A review of volunteered geographic information quality assessment methods. *Int. J. Geogr. Inf. Sci.* **31**(1), 139–167 (2017)
  180. Shahabi, C.: Towards a generic framework for trustworthy spatial crowdsourcing. In: Proceedings of the 12th International ACM Workshop on Data Engineering for Wireless and Mobile Access, pp. 1–4 (2013)
  181. She, J., Tong, Y., Chen, L.: Utility-aware social event-participant planning. In: Proceedings of the 2015 ACM International Conference on Management of Data, pp. 1629–1643 (2015)
  182. She, J., Tong, Y., Chen, L., Cao, C.C.: Conflict-aware event-participant arrangement. In: IEEE 31st International Conference on Data Engineering, pp. 735–746 (2015)
  183. She, J., Tong, Y., Chen, L., Cao, C.C.: Conflict-aware event-participant arrangement and its variant for online setting. *IEEE Trans. Knowl. Data Eng.* **28**(9), 2281–2295 (2016)
  184. Shen, W., Lopes, C.V., Crandall, J.W.: An online mechanism for ridesharing in autonomous mobility-on-demand systems. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence, pp. 475–481 (2016)
  185. Sheng, X., Tang, J., Zhang, W.: Energy-efficient collaborative sensing with mobile phones. In: 2012 IEEE Conference on Computer Communications, pp. 1916–1924 (2012)
  186. Song, T., Tong, Y., Wang, L., She, J., Yao, B., Chen, L., Xu, K.: Trichromatic online matching in real-time spatial crowdsourcing. In: 33rd IEEE International Conference on Data Engineering, pp. 1009–1020 (2017)
  187. Song, T., Xu, K., Li, J., Li, Y., Tong, Y.: Multi-skill aware task assignment in real-time spatial crowdsourcing. *GeoInformatica* (2019). <https://doi.org/10.1007/s10707-019-00351-4>
  188. Stevens, M., D’Hondt, E.: Crowdsourcing of pollution data using smartphones. In: Workshop on Ubiquitous Crowdsourcing (2010)
  189. Su, H., Zheng, K., Huang, J., Jeung, H., Chen, L., Zhou, X.: Crowdplanner: a crowd-based route recommendation system. In: 30th IEEE International Conference on Data Engineering, pp. 1144–1155 (2014)
  190. Sun, D., Xu, K., Cheng, H., Zhang, Y., Song, T., Liu, R., Xu, Y.: Online delivery route recommendation in spatial crowdsourcing. *World Wide Web* **11**, 1–22 (2018)
  191. Sweeney, L.: k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* **10**(5), 557–570 (2002)
  192. Tao, Q., Zeng, Y., Zhou, Z., Tong, Y., Chen, L., Xu, K.: Multi-worker-aware task planning in real-time spatial crowdsourcing. In: International Conference on Database Systems for Advanced Applications, pp. 301–317 (2018)
  193. Theodoridis, Y., Vazirgiannis, M., Sellis, T.K.: Spatio-temporal indexing for large multimedia applications. In: Proceedings of the IEEE International Conference on Multimedia Computing and Systems, pp. 441–448 (1996)
  194. Ting, H., Xiang, X.: Near optimal algorithms for online maximum edge-weighted b-matching and two-sided vertex-weighted b-matching. *Theor. Comput. Sci.* **607**, 247–256 (2015)
  195. To, H., Shahabi, C.: Location privacy in spatial crowdsourcing. In: Handbook of Mobile Data Privacy, pp. 167–194 (2018)
  196. To, H., Ghinita, G., Shahabi, C.: A framework for protecting worker location privacy in spatial crowdsourcing. *PVLDB* **7**(10), 919–930 (2014)
  197. To, H., Shahabi, C., Kazemi, L.: A server-assigned spatial crowdsourcing framework. *ACM Trans. Spat. Algorithms Syst.* **1**(1), 2 (2015)
  198. To, H., Asghari, M., Deng, D., Shahabi, C.: SCAWG: A toolbox for generating synthetic workload for spatial crowdsourcing. In: 2016 IEEE International Conference on Pervasive Computing and Communication Workshops, pp. 1–6 (2016)
  199. To, H., Fan, L., Tran, L., Shahabi, C.: Real-time task assignment in hyperlocal spatial crowdsourcing under budget constraints. In: IEEE International Conference on Pervasive Computing and Communications, pp. 1–8 (2016)
  200. To, H., Ghinita, G., Fan, L., Shahabi, C.: Differentially private location protection for worker datasets in spatial crowdsourcing. *IEEE Trans. Mob. Comput.* **16**(4), 934–949 (2017)
  201. To, H., Shahabi, C., Xiong, L.: Privacy-preserving online task assignment in spatial crowdsourcing with untrusted server. In: 34th IEEE International Conference on Data Engineering, pp. 833–844 (2018)
  202. Tong, X., Gupta, A., Lo, H., Choo, A., Gromala, D., Shaw, C.D.: Chasing lovely monsters in the wild, exploring players’ motivation and play patterns of pokémon go: go, gone or go away? In: Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, Companion Volume, pp. 327–330 (2017)
  203. Tong, Y., Zhou, Z.: Dynamic task assignment in spatial crowdsourcing. In: Proceedings of the 26rd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, vol. 10, no. 2, pp. 18–25 (2018)
  204. Tong, Y., She, J., Ding, B., Chen, L., Wo, T., Xu, K.: Online minimum matching in real-time spatial data: experiments and analysis. *PVLDB* **9**(12), 1053–1064 (2016)
  205. Tong, Y., She, J., Ding, B., Wang, L., Chen, L.: Online mobile micro-task allocation in spatial crowdsourcing. In: 32nd IEEE International Conference on Data Engineering, pp. 49–60 (2016)
  206. Tong, Y., Chen, L., Shahabi, C.: Spatial crowdsourcing: challenges, techniques, and applications. *PVLDB* **10**(12), 1988–1991 (2017)
  207. Tong, Y., Chen, Y., Zhou, Z., Chen, L., Wang, J., Yang, Q., Ye, J., Lv, W.: The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1653–1662 (2017)
  208. Tong, Y., Wang, L., Zhou, Z., Ding, B., Chen, L., Ye, J., Xu, K.: Flexible online task assignment in real-time spatial data. *PVLDB* **10**(11), 1334–1345 (2017)
  209. Tong, Y., Chen, L., Zhou, Z., Jagadish, H.V., Shou, L., Lv, W.: SLADE: a smart large-scale task decomposer in crowdsourcing. *IEEE Trans. Knowl. Data Eng.* **30**(8), 1588–1601 (2018)
  210. Tong, Y., Wang, L., Zhou, Z., Chen, L., Du, B., Ye, J.: Dynamic pricing in spatial crowdsourcing: a matching-based approach. In: Proceedings of the 2018 ACM International Conference on Management of Data, pp. 773–788 (2018)

211. Tong, Y., Zeng, Y., Zhou, Z., Chen, L., Ye, J., Xu, K.: A unified approach to route planning for shared mobility. *PVLDB* **11**(11), 1633–1646 (2018)
212. Tran, L., To, H., Fan, L., Shahabi, C.: A real-time framework for task assignment in hyperlocal spatial crowdsourcing. *ACM Trans. Intell. Syst. Technol.* **9**(3), 37:1–37:26 (2018)
213. U, L.H., Yiu, M.L., Mouratidis, K., Mamoulis, N.: Capacity constrained assignment in spatial databases. In: *Proceedings of the 2008 ACM International Conference on Management of Data*, pp. 15–28 (2008)
214. Vansteenwegen, P., Souffriau, W., Oudheusden, D.V.: The orienteering problem: a survey. *Eur. J. Oper. Res.* **209**(1), 1–10 (2011)
215. Venanzi, M., Guiver, J., Kazai, G., Kohli, P., Shokouhi, M.: Community-based Bayesian aggregation models for crowdsourcing. In: *Proceedings of the 23rd International Conference on World Wide Web*, pp. 155–164 (2014)
216. Vu, K., Zheng, R., Gao, J.: Efficient algorithms for k-anonymous location privacy in participatory sensing. In: *Proceedings of the IEEE International Conference on Computer Communications*, pp. 2399–2407 (2012)
217. Wang, L., Zhang, D., Yang, D., Lim, B.Y., Ma, X.: Differential location privacy for sparse mobile crowdsensing. In: *IEEE 16th International Conference on Data Mining*, pp. 1257–1262 (2016)
218. Wang, Q., He, W., Wang, X., Cui, L.: Quality-assure and budget-aware task assignment for spatial crowdsourcing. In: *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pp. 60–70 (2016)
219. Wang, Q., He, W., Wang, X., Cui, L.: Quality-assure and budget-aware task assignment for spatial crowdsourcing. In: *12th International Conference on Collaborate Computing: Networking, Applications and Worksharing*, pp. 60–70 (2016)
220. Wang, Y., Wong, S.C.: Two-sided online bipartite matching and vertex cover: beating the greedy algorithm. In: *42nd International Colloquium on Automata, Languages, and Programming*, pp. 1070–1081 (2015)
221. Wang, Y., Tong, Y., Long, C., Xu, P., Xu, K., Lv, W.: Adaptive dynamic bipartite graph matching: a reinforcement learning approach. In: *35th IEEE International Conference on Data Engineering*, pp. 1478–1489 (2019)
222. Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., Movellan, J.R.: Whose vote should count more: optimal integration of labels from labelers of unknown expertise. In: *Advances in Neural Information Processing Systems*, pp. 2035–2043 (2009)
223. Williamson, D.P., Shmoys, D.B.: *The Design of Approximation Algorithms*. Cambridge University Press, Cambridge (2011)
224. Wong, R.C., Tao, Y., Fu, A.W., Xiao, X.: On efficient spatial matching. In: *Proceedings of the 33rd International Conference on Very Large Data Bases*, pp. 579–590 (2007)
225. Wu, P., Ngai, E.W., Wu, Y.: Toward a real-time and budget-aware task package allocation in spatial crowdsourcing. *Decis. Support Syst.* **110**, 107–117 (2018)
226. Xia, H., Yang, H.: Is last-mile delivery a ‘killer app’ for self-driving vehicles? *Commun. ACM* **61**(11), 70–75 (2018)
227. Xu, Y., Tong, Y., Shi, Y., Tao, Q., Xu, K., Li, W.: An efficient insertion operator in dynamic ridesharing services. In: *35th IEEE International Conference on Data Engineering*, pp. 1022–1033 (2019)
228. Yang, C., Lin, K.: An index structure for improving closest pairs and related join queries in spatial databases. In: *International Database Engineering & Applications Symposium*, pp. 140–149 (2002)
229. Yao, A.C.: How to generate and exchange secrets (extended abstract). In: *27th Annual Symposium on Foundations of Computer Science*, pp. 162–167 (1986)
230. Yu, H., Miao, C., Shen, Z., Leung, C.: Quality and budget aware task allocation for spatial crowdsourcing. In: *Proceedings of the 14th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1689–1690 (2015)
231. Yuan, J., Zheng, Y., Xie, X., Sun, G.: T-drive: enhancing driving directions with taxi drivers’ intelligence. *IEEE Trans. Knowl. Data Eng.* **25**(1), 220–232 (2013)
232. Zeng, Y., Tong, Y., Chen, L., Zhou, Z.: Latency-oriented task completion via spatial crowdsourcing. In: *34th IEEE International Conference on Data Engineering*, pp. 317–328 (2018)
233. Zhai, D., Sun, Y., Liu, A., Li, Z., Liu, G., Zhao, L., Zheng, K.: Towards secure and truthful task assignment in spatial crowdsourcing. *World Wide Web* **22**, 2017–2040 (2018)
234. Zhang, C.J., Tong, Y., Chen, L.: Where to: crowd-aided path selection. *PVLDB* **7**(14), 2005–2016 (2014)
235. Zhang, G., Zhu, A., Huang, Z., Ren, G., Qin, C., Xiao, W.: Validity of historical volunteered geographic information: evaluating citizen data for mapping historical geographic phenomena. *Trans. GIS* **22**(1), 149–164 (2018)
236. Zhang, J., Wen, D., Zeng, S.: A discounted trade reduction mechanism for dynamic ridesharing pricing. *IEEE Trans. Intell. Transp. Syst.* **17**(6), 1586–1595 (2016)
237. Zhang, L., Hu, T., Min, Y., Wu, G., Zhang, J., Feng, P., Gong, P., Ye, J.: A taxi order dispatch model based on combinatorial optimization. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2151–2159 (2017)
238. Zhang, X., Yang, Z., Zhou, Z., Cai, H., Chen, L., Li, X.: Free market of crowdsourcing: incentive mechanism design for mobile sensing. *IEEE Trans. Parallel Distrib. Syst.* **25**(12), 3190–3200 (2014)
239. Zhang, X., Yang, Z., Sun, W., Liu, Y., Tang, S., Xing, K., Mao, X.: Incentives for mobile crowd sensing: a survey. *IEEE Commun. Surv. Tutor.* **18**(1), 54–67 (2016)
240. Zhang, X., Yang, Z., Liu, Y., Tang, S.: On reliable task assignment for spatial crowdsourcing. *IEEE Trans. Emerg. Top. Comput.* **7**(1), 174–186 (2019)
241. Zhang, Y., Chen, Q., Zhong, S.: Privacy-preserving data aggregation in mobile phone sensing. *IEEE Trans. Inf. Forensics Secur.* **11**(5), 980–992 (2016)
242. Zhao, B., Xu, P., Shi, Y., Tong, Y., Zhou, Z., Zeng, Y.: Preference-aware task assignment in on-demand taxi dispatching: An online stable matching approach. In: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pp. 2245–2252 (2019)
243. Zhao, D., Li, X., Ma, H.: How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint. In: *Proceedings of the IEEE Conference on Computer Communications*, pp. 1213–1221 (2014)
244. Zhao, Y., Han, Q.: Spatial crowdsourcing: current state and future directions. *IEEE Commun. Mag.* **54**(7), 102–107 (2016)
245. Zhao, Y., Li, Y., Wang, Y., Su, H., Zheng, K.: Destination-aware task assignment in spatial crowdsourcing. In: *Proceedings of the 2017 ACM Conference on Information and Knowledge Management*, pp. 297–306 (2017)
246. Zhao, Z., Wei, F., Zhou, M., Chen, W., Ng, W.: Crowd-selection query processing in crowdsourcing databases: A task-driven approach. In: *Proceedings of the 18th International Conference on Extending Database Technology*, pp. 397–408 (2015)
247. Zheng, L., Chen, L., Ye, J.: Order dispatch in price-aware ridesharing. *PVLDB* **11**(8), 853–865 (2018)
248. Zheng, Y., Li, G., Li, Y., Shan, C., Cheng, R.: Truth inference in crowdsourcing: is the problem solved? *PVLDB* **10**(5), 541–552 (2017)