A survey of spatial crowdsourcing

Gummidi, Srinivasa Raghavendra Bhuvan; Xie, Xike; Pedersen, Torben Bach

Published in:
ACM Transactions on Database Systems

DOI (link to publication from Publisher):
10.1145/3291933

Publication date:
2019

Document Version
Accepted author manuscript, peer reviewed version

Citation for published version (APA):
Gummidi, S. R. B., Xie, X., & Pedersen, T. B. (2019). A survey of spatial crowdsourcing. *ACM Transactions on Database Systems*, *44*(2), 8:1-8:46. Article 8. https://doi.org/10.1145/3291933

A Survey of Spatial Crowdsourcing

ANONYMOUS AUTHOR(S)

Widespread usage of advanced mobile devices has led to the emergence of a new class of crowdsourcing called spatial crowdsourcing. Spatial crowdsourcing advances the potential of a crowd to perform tasks related to real-world scenarios involving physical locations, which were not feasible with conventional crowdsourcing methods. The main feature of spatial crowdsourcing is the presence of spatial tasks that require workers to be physically present at a particular location for the task fulfillment. Research related to this new paradigm has gained momentum in recent years, thus necessitating a comprehensive survey to offer a bird's eye view of the current state of spatial crowdsourcing literature. In this paper, we discuss the spatial crowdsourcing infrastructure and identify the fundamental differences between spatial and conventional crowdsourcing. Furthermore, we provide a comprehensive view of the existing literature by introducing a taxonomy, elucidate the issues/challenges faced by different components of spatial crowdsourcing, and suggest potential research directions for the future.

CCS Concepts: • **Information systems** → **Geographic information systems**; • **Human-centered computing** → **Collaborative and social computing**; • **Security and privacy** → Privacy protections;

Additional Key Words and Phrases: Algorithms, Task Scheduling, Spatial Crowdsourcing, Task assignment, Task Matching, Rewards, Incentive Mechanism, Quality Assurance, Location Privacy, Spatial Databases

ACM Reference Format:
Anonymous Author(s). 1. A Survey of Spatial Crowdsourcing. *ACM Trans. Datab. Syst.* 1, 1, Article 1 (January 1), 45 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

The proliferation of advanced mobile devices opened up a new crowdsourcing avenue (spatial crowdsourcing), to harness the potential of the crowd to perform real-world tasks with a strong spatial nature, which are not supported by conventional crowdsourcing (CC) techniques. CC techniques [9] focus on transactions that are carried out entirely through the medium of the Internet. However, in the real-world crowdsourcing scenarios such as crowdsourcing disaster response [138] and news reporting [116], tasks are likely to have spatial requirements that cannot be fulfilled virtually and require physical on-location operations. Spatial crowdsourcing (SC) is a particular class of crowdsourcing; that deals with such spatial tasks.

The phrase "Spatial Crowdsourcing" was introduced in 2012 by Kazemi et al [61], though the technique was already in use for some years. The eBird[1] project [99] is one of the earliest SC-based applications for collecting information about bird sightings from the bird-watching enthusiasts. SC is also referred in the literature by the following phrases: "Place-Centric Crowdsourcing" [21, 59, 114], "Location-based Crowdsourcing" [2, 7, 115], "Participatory Sensing" [10, 32, 40], "Mobile Crowdsourcing" [11, 125], and "Mobile Crowdsensing" [44, 122]. To elaborate, "Participatory Sensing" and "mobile crowdsensing" belong to a subset of SC, that involves utilization of smartphones of

[1] https://ebird.org

0362-5915/1/1-ART1 $15.00
https://doi.org/0000001.0000001

users to collect the different sensor data of the smartphones at different locations. In general, a typical participatory sensing or mobile crowdsensing application does not harness the wisdom or knowledge of the crowd. However, SC offers avenues to utilize both the sensor information as well as the knowledge possessed by the humans. A participant's skillset can be considered during recruitment for performing a given job or task. Furthermore, in SC, the human/participant effort is often associated with constraints such as spatiotemporal availabilities, capabilities, and rewards. Such limitations make human knowledge and common sense a valuable resource that should be carefully planned than in the case of mobile crowdsensing. Therefore, techniques for effective utilization of human knowledge are desired and developed, like task matching and subsequent scheduling of tasks agreed by the participant. For example, consider the SC application for collecting traffic information at different spots in the city, as described below.

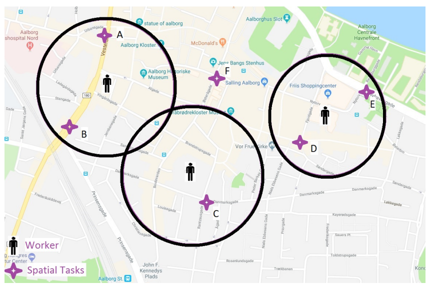


Fig. 1. Spatial Crowdsourcing: Performing different traffic data collection tasks in the city

Spatial Crowdsourcing Example: Fig. 1 depicts an SC example of collecting traffic information at different points in the city, with the help of the crowd. The figure shows three participants with neighbourhood boundaries depicted by the black circles around the participant location, along with the different neighbouring tasks. The participants can only collect the traffic information at the task locations that lie in their neighbourhoods. The tasks involve harnessing the human knowledge to retrieve qualitative details, utilization of the sensors in the participant's smartphone, or both. For instance, tasks A & E involve capturing pictures of the road at a particular time. Tasks B & C consist of estimating the volume of traffic that passes through that point in a 5-minute interval during a predefined period of the day. Tasks D & F requires inputs from participants to verify whether a traffic jam is typical during a particular time of the day at those task locations. The SC application attempts to match the participants with the tasks based on their availabilities and constraints. Moreover, the SC application offers to schedule the tasks that are agreed by the participant.

Typically, the tasks involved in SC require movement of the contributor or worker to the tasks' locations, to perform the task. The task's requester would specify the task's location along with the necessary details like task completion deadlines, task descriptions, associated rewards, skills, and reputation expected from the worker. SC has the potential for collecting information for a broad range of applications in domains like environmental data collection (NoiseTube platform [98]), transportation (Uber[2]), journalism [69, 70], and business intelligence (Gigwalk[3] and TaskRabbit[4]). In addition to the data collection and query answering tasks, SC can be extended to service complex spatial tasks like employing a set of workers with diverse skills to renovate the house, hiring workers to perform household chores, etc.

[2] https://www.uber.com
[3] http://www.gigwalk.com/
[4] https://www.taskrabbit.com/

Recently, research on SC has gained momentum; consequently, many techniques are proposed for various application scenarios. Therefore, compiling and organizing the existing research regarding SC will be beneficial for researchers to gain a comprehensive view of current research and potential directions for future studies. Although there exist surveys of conventional crowdsourcing like [20, 71, 80, 128], none of them delves into the topic of SC in detail. Recently, a short article [134] briefly discussed the current state and existing challenges. However, [134] does not offer a technical perspective and a comparison between the existing strategies in SC. Similarly, a tutorial reviewing the current state of SC research was published in August 2017 [109]. In comparison, our survey additionally offers a comprehensive analysis of the relationships between different types of constraints and compares different optimization problems in detail and highlight the limitations. Furthermore, our survey classifies the spatial tasks, and discusses the truth inference models and data aggregation methods in SC. Similarly, the survey [41] presents an overview of existing applications on crowdsensing until mid-2011. It does not address technical details such as task matching, assignment, and scheduling which are crucial for SC data management. Recently, the survey [90] has summarized the main aspects of Quality of Information (QoI) in mobile crowdsensing. However, the survey only deals with the estimation methods of QoI present in the current mobile crowdsensing literature. The present paper provides all these contributions and aims to improve the reviews above by performing a comprehensive review of SC by surveying the research done in the area until May 2018. This paper mainly focuses on the techniques and their comparisons along with the applications pertaining specifically to SC. We provided an overview of the different problems in SC as well as how different constraints of SC impact each other. This paper summarizes all the major aspects of SC, however it does not delve into the security and privacy aspects in detail like the survey [37] which focuses only on the security, privacy and trust aspects of mobile crowdsourcing.

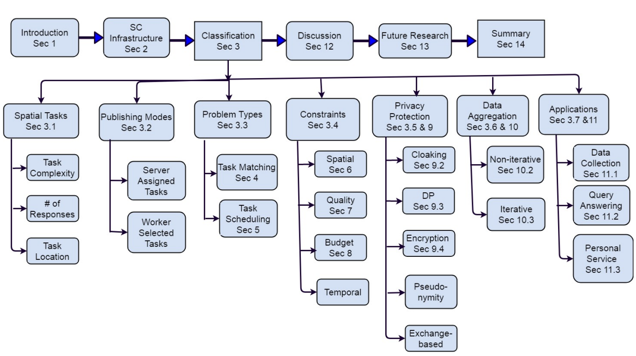


Fig. 2. Structure of the Paper

This paper is organized into different sections as shown in Fig. 2. *Section 2: Spatial Crowdsourcing Infrastructure* describes in detail the unique features of SC relative to CC concerning the building blocks of crowdsourcing. *Section 3: Classification* introduces our taxonomy for organizing the existing literature into different broad categories based on various criteria like types of spatial tasks, task publishing modes, type of optimization problems, addressed constraints, privacy protection techniques, data aggregation techniques, and type of applications. As can be seen in Fig. 2, the part on classification contains the majority of the material in the paper and would have led to deep nesting in its hierarchy. To improve the readability, we have instead decided to flatten the hierarchy by discussing the prominent sub-categories in separate sections.

Accordingly, the two types of problems in SC are discussed in *Section 4: Task Matching Problem* and *Section 5: Task Scheduling Problem*, respectively. Constraints that affect the setting of such problems are discussed in *Section 6: Spatial Constraints*, *Section 7: Quality Constraints*, and *Section 8: Budget Constraints*, respectively[5]. Additionally, a comparison is performed between different techniques of SC and the limitations are identified, based on the association among various constraints.*Section 9* details the different privacy protection techniques and the impact of privacy protection on task matching problems. *Section 10* describes the data aggregation techniques that are commonly employed for truth detection in SC. *Section 11: Applications* details three common types of SC applications. The challenges/issues posed in the reviewed SC literature are discussed in *Section 12: Discussion*. Finally, in *Section 13: Future Research Directions* and *Section 14: Summary*, we provide potential research directions and the summary.

2 SPATIAL CROWDSOURCING INFRASTRUCTURE

Spatial crowdsourcing infrastructure comprises of four major components: requester, spatial task, worker, and SC server as shown in Fig. 3. This section discusses each SC infrastructure component in detail and outlines the differences concerning the core components in both CC [49] and SC [61] systems.

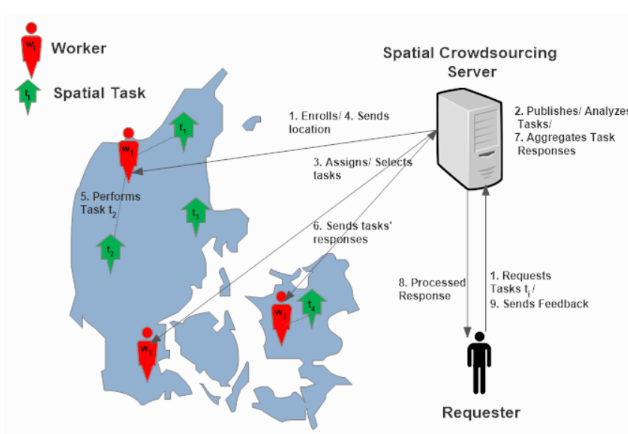


Fig. 3. Spatial Crowdsourcing Workflow Scenario

2.1 Requester

A requester is a real-world entity like a person or an organization that requests for a particular spatial task to be completed by the crowd at a specific location (see Fig. 3). Similar to the CC, the requester is the initiating point in the SC process. A requester designs the task and sets the conditions that need to be satisfied for performing the task. The requester has certain responsibilities such as accepting the answers provided or data collected by the crowd and giving them feedback [14]. The role of the requester is the same in SC and CC, except for the geospatial knowledge required while specifying the spatial tasks and the responsibility to respect the privacy of the worker by not exploiting the worker's task location visit information.

2.2 Worker

A worker is a person that participates in the process of SC, with an objective to perform the assigned/selected spatial task. Though "worker" is a commonly used term to refer these contributors, we use the nomenclature of referring worker as worker, to be in line with the phrase "requester".

[5]Temporal constraints are minor issues in the context of SC, and are not discussed in detail.

The main difference between the features of the worker mentioned in the CC system and the SC is the motivation to **physically move** to a particular geographical location to perform a spatial task.
Worker Definition: There were different definitions mentioned in the literature for a worker W, like:

$$W =< l, R, maxT, Re, E, \theta > \tag{1}$$

Most of the definitions essentially had the following basic and optional attributes:
Basic attributes [61]

- Current physical location of the worker l
- Region of interest R
- Maximum number of tasks $maxT$

Optional attributes:

- Expected reward Re [27]
- Skillset E [27]
- Worker's reputation score calculated by SC-server θ [62]

2.3 Spatial task

As discussed earlier, a spatial task is fundamentally different from the CC task due to the condition that the worker should be physically present at the task's location. A spatial task is requested by the requester and is fulfilled by the worker. It is a location-specific activity like answering a question about a local restaurant, taking pictures of a local tourist spot, or collecting noise pollution data. Moreover, a SC task will be defined by the requester with a set of requirements for assigning or allowing a person from the crowd to work on the task. The requirements might include the minimum level of skills required to fulfil a task, the number of answers needed, the expertise of the user, the deadline before which the task needs to be completed, and their experience in solving similar tasks.
Spatial Task Definition: There were different definitions mentioned in the literature for a spatial task t, like:

$$t =< l, q, ti_{\mathrm{i}}, ti_{\mathrm{e}}, r, \alpha, E, Cat > \tag{2}$$

Most of the definitions essentially had the following basic and optional attributes:
Basic attributes [61]

- Physical location of the task l
- Query description q
- Issuing time ti_{i}
- Expiration time ti_{e}

Optional attributes:

- Associated reward r [27]
- Minimum threshold for worker's reputation α [62]
- Expected worker's skillset E [15]
- Type/category of task Cat [47]
- Maximum number of workers $maxW$ [132]

2.4 Spatial Crowdsourcing Server

Through the SC-server, the requester requests a task to be performed by the crowd, the workers register to be assigned or select the tasks. It delegates the communication between the requester and the worker of the spatial task, facilitates the process to satisfy the task requirements, assigns tasks to workers based on location, helps improving the quality of the outcome by executing different strategies, identifies the anomalies and detects fraudulent responses, and protects the privacy of the involved stakeholders. Furthermore, there are additional functionalities for SC related to communication between different stakeholders.

Communication between different stakeholders: SC-server enables the broadcast of information between the requesters who request tasks and the crowd [14, 49]. According to [49], there are four major features for a CC platform: “Crowd-related interactions”, “Requester-related interactions”, “Task-related interactions”, and “Platform-related facilities”. This categorization is relevant for the SC paradigm as well. In addition to the features mentioned before in [49], some additional features need to be adapted in the case of SC. The adaptations are:

a. For the Crowd

- The power to disclose or hide their locations to the SC-server.
- Ability to select tasks that need to be performed near their location.
- Ability to specify a spatial region in which they wish to work, i.e., the geographical extent they can travel for performing a spatial task.
- Ability to provide location privacy-protection of workers.
- Ability to reject assigned tasks.

b. For Requested Tasks

- Determining the maximum acceptable distance of the workers who are in the near vicinity of the task.
- Defining the geographical location where the requested task needs to be performed.
- Ability to hide the location where the task needs to be performed or only to be shown to workers who accepted or selected the task.

Spatial Task Lifecycle: Fig. 3 illustrates the lifecycle of a typical SC task. The lifecycle of a spatial task begins with the requester requesting the SC-server to facilitate the fulfilment of specified spatial tasks (Step 1 of Fig. 3). The SC-server publishes the requested tasks actively by pushing them to selected workers or passively by publishing them as a list from which the workers can choose based on their interests (Steps 2 & 3 of Fig. 3). The worker performs the assigned/chosen task and responds to the SC-server with the collected information/answer for the task query (Steps 5 & 6 of Fig. 3). The SC-server processes/aggregates the responses from workers and delegates them to the requesters (Steps 7 & 8 of Fig. 3). The requesters validate the task responses and provide feedback regarding the quality of the task response (Step 9 of Fig. 3). The workflow is similar to the lifecycle of a CC task, except for the consideration of spatial characteristics like travel time/distance between potential workers and spatial tasks, and the spatiotemporal behaviour of workers.

3 CLASSIFICATION

The current literature in SC can be classified into distinct categories based on the following criteria: the type of problem being addressed, the modes of publishing spatial tasks, the different constraints being considered, the type of application, and the type of spatial tasks. This section provides a brief overview of the different categories and elaborate them in detail in the following sections (see Table 1).

3.1 Spatial Tasks

Similar to CC tasks, we can classify spatial tasks based on their complexity and the required number of workers. Additionally, we can categorize spatial tasks by the spatial extent of the task's

[6] www.mturk.com

Table 1. Classification of Spatial Crowdsourcing based on different constraints

		Spatial Task Publishing Modes			
		Server Assigned Tasks		Worker Selected Tasks	
Constraints	Spatial constraints	a. Spatial Region [26, 61, 62] b. Direction of worker's Commute [12, 17]		Spatial Region [29, 73]	
	Quality Constraints	a. Worker Reputation: [17, 62, 88] b. Worker Expertise: [15, 27, 107] c. Truth Inference [45, 50, 85]		Qualification tests [76, 92]	
	Temporal Constraints	a. Deadline of Task [61] b. Delivery Task PickUp time and Deadline [100] c.Deadline for Destination [73]		Deadline for task [29]	
	Budget Constraints	Reward models and Incentive Mechanisms [27, 38, 48, 64, 127]		Fixed Rewards for Tasks[6] [2]	
Scenarios		Online	Offline	Online	Offline
Types of Problems	Task Matching	MAB [47], GOMA [111], f-MTC, d-MTC [103, 113], FTOA [112], TOM [97]	MTA [61], MSA [107], MTMCA [27], RDB-SC [17], MRA [133], PAPA [60], DPTA [105], MS-SC [15]	-NA-	-NA-
	Task Scheduling	GALS [30], TRACCS [12]	MTS [29]	OnlineRR [73], Auction-SC [5]	MTS [29]
Privacy Protection		a. Differential Privacy-based [104, 105, 131] b. Spatial Cloaking [60, 87] c. Encryption-based [95, 96]		a. Pseudonymity [22] b. Exchange-based [130]	

location.

Complexity of the spatial task: The spatial tasks are classified into two types based on their complexity.

- **Atomic tasks**: An atomic task, as the name implies, cannot be divided down into sub-tasks. Atomic tasks are simple and can be performed by one worker [61]. An example of an atomic task is to answer yes or no to the query "Is it raining now?".
- **Complex tasks**: A complex task consists of two or more related activities that need to performed collaboratively for accomplishing the spatial task. These complex tasks will be divided into atomic sub-tasks that require specific skills [7, 15, 61, 107]. For instance, a complex task like renovating a shop at a particular location involves multiple activities like procuring materials, performing repairs, and painting.

Number of responses: A spatial task can be assigned to either one or more workers. By assigning to multiple workers, the quality of the outcome can be improved by being able to assess the majority answers. However, it is not certain to receive responses regarding the assigned tasks from the workers; they can either reject or ignore the tasks. In case the worker does not reply; then the server will wait till the expiration time of the spatial task [47]. A spatial task can be classified into two types based on the number of responses, single and multiple responses.

Task's Physical location: The task location could be a particular point, a line segment or a region in geographical space.(See Fig. 4)

- **Point task**: The spatial task is required to be performed at that particular point location [51, 61, 62]. For example, a worker has to visit the point of interest A (As seen in Fig. 4) for performing a task like enquiring the day's menu of a university cafeteria, where the worker has to take a picture of the menu details of the cafeteria to complete the task.
- **Area/Region task**: The spatial task is required to be performed by the worker in a region of geographical space, instead of a particular location [103]. For example, a worker has to visit

Fig. 4. Spatial task type based on task's location condition

the building B in the university campus for collecting noise pollution data of a building. The task can be carried out by collecting information at any point in the building.
- **Delivery Task**: The spatial task has two parts: pick up the package from the point of interest A and deliver the package to the building B [100]. For example, collecting a food package from the restaurant at location A and delivering it to the customer located at B.

3.2 Modes of Publishing Spatial Tasks

There are two ways of publishing spatial tasks [61] on the SC-server:

- **Server Assigned Tasks**: The SC-server chooses available suitable workers for a given spatial task based on different parameters like their proximity to the spatial task, availability to perform the task, abilities to match the requirements of the task, reliability of the worker to assure some degree of quality in task outcomes.
- **Worker Selected Tasks**: Workers selects spatial tasks of their interest from a given list published by the SC-server.

The two task publishing modes, Server Assigned Tasks (SAT) and Worker Selected Tasks (WST) are discussed implicitly along with the other classification categories, as every work in SC follows either one of them. Furthermore, most of the existing work is dominated by the SAT task publishing mode as the SC-server can exert control in this mode by pushing tasks to workers. On the contrary, in the WST mode workers select the tasks based on their interest with little or no influence from the SC-server. Therefore, the SAT task publishing mode is discussed more in detail when compared to WST mode.

3.3 Types of Problems

We can classify the numerous optimization problems that are addressed in the SC literature into two different categories: Task Matching and Task Scheduling.

- **Task Matching**: Given a set of spatial tasks T and a set of workers W, the SC-server tries to match the workers to the spatial tasks. This problem exists only when the SC-server publishes the task through *Server Assigned Tasks* mode [61] (discussed in Section 4).
- **Task Scheduling**: Given a set of tasks that are selected by the worker [29] or assigned to a particular worker [30], the Task Scheduling problem tries to schedule the order of tasks to achieve different optimization goals like maximizing the number of tasks completed by a worker, maximizing the rewards received by the workers, etc. The problem of task scheduling is unique to SC (discussed in Section 5).

Besides, the algorithms proposed for solving the problems of task matching and task scheduling require knowledge of the inputs like worker's location at different times of the day and tasks information. Based on the knowledge of the inputs we can categorize them into two scenarios (discussed in Section 4 and 5):

- **Offline Scenario**: In this scenario, the information about the inputs is available to the SC-server from the beginning, i.e., the algorithm has the complete knowledge about the inputs that arrive at a future time. In this scenario, the algorithms can maximize the global objective functions like maximizing the number of tasks assigned [61]. This scenario, though not effective in real-world case scenarios of SC, would help understand the difficulties in the online scenario, due to the uncertainty of inputs arrival to the system. This scenario is effective in certain special cases where the schedule of the worker and the arrival of tasks are known beforehand to the system.
- **Online Scenario**: In this scenario, the algorithm receives inputs in a streaming fashion; and thus, the algorithm has to process each input immediately [58]. This scenario represents the majority real-world case scenarios of SC, where the workers and tasks are dynamic [5, 97, 110–112], i.e., their arrival orders are not known beforehand.

Optimization goals: Typically, task matching and task scheduling problems described in the SC literature optimize certain aspects like maximizing the number of tasks assigned. The optimization goals can be categorized from either the perspective of workers or the SC-server [134] (discussed in Section 4 and Section 5):

- **Worker's Perspective**: From a worker's perspective, the goal would be to maximize the net reward that she receives from the SC-server for performing the assigned or selected tasks [111]. Generally, the net reward is the difference between the reward associated with the task and the travel cost incurred to travel to the location of the task. In the case of voluntary SC, the worker's goal would be to minimize the total travelling distance.
- **SC-server's Perspective**: From the SC-server's perspective. the goal would be to maximize the number of assigned tasks [61], maximize the number of accepted tasks [105], maximize the quality score [17, 107], minimize the incentives paid to the worker [27], minimize the total travel costs of all workers [30], maximize the perfect matching [35] with no unstable pairs [121] and maximize the number of tasks completed by worker before deadline [29].

3.4 Constraints

In an ideal situation, the spatial tasks can be assigned to all the workers to retrieve better results. However, this is not the case in reality, due to the different constraints of the workers like preferred region of work, preferred task types, and the travelling cost taken for performing the task. We can classify the various constraints broadly into the following four categories:

- **Temporal Constraints**: relate to the deadline of the task [61], or the time duration during which the worker is available for work [73]. Temporal constraints are common CC constraints; hence they are not further detailed.
- **Spatial Constraints**: relate to the spatial preferences of a worker or a task, like the preferred locations where she likes to work [61], the maximum travel distance she is willing to travel to perform tasks [12], and the preferred region within which the task should be performed [103] (discussed in detail in Section 6).
- **Quality Constraints**: refer to the expected quality of the responses from the workers after performing the task. They usually involve pre-task qualification tests [92] or assignment based on previous histories [62]/abilities [27] of the worker (discussed in detail in Section 7).
- **Budget Constraints**: refer to the restrictions on the incentive mechanism in rewards-based SC, where reward will be offered to workers for performing the task. Some of the constraints

are worker's expected reward or total threshold reward for the requester's requested tasks (discussed in detail in Section 8).

3.5 Privacy Protection

Due to the sensitive nature of the worker's location, preserving location privacy is the responsibility of the SC-server. The worker might not trust the SC-server to share her physical location. Therefore, workarounds are proposed to address the privacy concerns like using the concept of differential privacy [105]. The privacy-preserving techniques can be categorized into the following five types (discussed in detail in Section 9):

- **Pseudonymity Techniques**: uncouple the person's identity and the submitted data [22].
- **Cloaking Techniques**: hides the exact locations of the workers in a cloaked region [60, 87].
- **Exchange-based Techniques**: exchange the crowdsourced information among the workers before disclosing it to the untrusted SC-server, to obscure the individual workers. [130]
- **Encryption-based Techniques**: hide the identity and the location of the workers from the SC-server [95, 96]
- **Differential Privacy-based Techniques**: distort the workers location information by adding artificial noise [104, 105, 131]

3.6 Crowdsourced Data Aggregation

In many cases, tasks in SC requires multiple responses from the workers, for example for ascertaining the traffic situation of an area. However, the multiple responses from workers may be both agreeing and conflicting. To establish the truth, the crowdsourced data needs to be aggregated and relayed back to the task requester as a single value. There are numerous methods in CC for crowdsourced data aggregation, categorized based on their computing model [53]:

- ***Non-iterative Aggregation***: aggregates the responses for each question to a single value. Examples are Majority Voting [65], Honeypot [68], and Expert Label Injected Crowd Estimation(ELICE) [63].
- ***Iterative Aggregation***: calculates the aggregated value iteratively for each question based on the expertise of the workers who answered and updates the worker expertise iteratively based on the answer given by the worker. Examples are Expectation Maximization (EM) [54], Generative Model of Labels, Abilities, and Difficulties (GLAD) [120], Supervised Learning from Multiple Experts(SLME) [89], and Iterative Learning(ITER) [57].

For truth inference, SC extends the above-mentioned data aggregation methods by considering different spatial attributes like distance to the task from the worker's location [50], task location popularity/influence, location visit tendencies [85]. These methods are discussed in detail in Section 10.

3.7 SC Applications

Many applications are developed implementing the methods proposed in SC-literature, for serving different purposes like collecting data during disasters, environmental data collection, delivering packages, and ride sharing services. We can broadly classify the existing applications into three categories based on the nature of human involvement, data collection, query answering, and Personal Service. The three common types of applications are discussed in detail in Section 11.

The remaining sections are organized based on the *type of problems* and the different *constraints* considered. We have chosen this organisation instead of organizing the sections according to the classification schema, as the latter would have led to massive redundancy among sections. For example, the majority of the SC literature falls within the two types of task publishing modes. Therefore, they are discussed implicitly as part of each section to avoid redundancy.

4 TASK MATCHING PROBLEM

4.1 Introduction

As discussed earlier, the task matching problem involves assigning a given set of tasks to the suitable workers based on various conditions. The problem exists exclusively for the *Server Assigned Tasks* publishing mode as the *Worker Selected Tasks* publishing mode does not require the SC-server to choose the workers [61]. Typically, a task matching problem aims to achieve optimization goals benefitting the SC-server. For instance, maximizing the number of tasks assigned [61, 107], minimizing the cost incurred by the server [26, 113], improving the quality of task responses [62] or goals benefitting the workers like maximizing the reward received by the worker [12].

To define the task matching/assignment problem, let us say that at a given time instance, there are a set of tasks ($t_1, t_2, t_3, \ldots$) requested to the SC-server by the requesters [107]. These tasks consist of the task/query (q) that needs to be performed at location (l) before the deadline. These tasks will be assigned to the workers available at the particular time instance. The available workers are determined through the "Task Inquiries" [107] made by the workers. Through these *Task Inquiries*, workers (W_1, W_2, W_3,...) share their locations l_i to the SC-server along with their constraints like spatial region and the maximum number of tasks that can be performed per time instance. The task assignment set contains the pairs of worker-spatial task matches($< W, t >$) at time instance s.

4.2 Offline Scenario: Known Arrival order of Tasks and Workers

As mentioned in Section 3.3, in the offline scenario the SC-server has complete knowledge about the inputs that arrive at a future time, in this case, the tasks' and workers' arrival order. Given a set of workers and tasks, the ideal way of solving the matching problem is to assign all the workers to all the tasks. However, in reality, it is not possible owing to the different constraints of the spatial task and the workers. For instance, one would end up with a scenario where workers are required to travel long distances just to solve the task, which is not likely to be accepted by the worker to perform the spatial task. However, a higher number of assignments results in solving a greater number of spatial tasks requested by the requesters. This leads to an optimization problem of **maximizing the number of task assignments** by the SC-server("Maximum Task Assignment Problem") [61]. Similarly, some of the other optimization problems according to different optimization objectives can be defined as:

- Maximum Score Assignment Problem [107]: The objective is to maximize the overall worker expertise score that results in higher quality results, where $score(w, t)$ is the value indicating the compatibility between worker w and task t based on worker expertise.
- Maximum Task Minimum Cost Assignment [27]: The objective is to maximize the number of expert matches while minimizing the total cost paid to workers.
- Maximum Task Scheduling with Multiple Workers [30]: The objective is to maximize the number of completed tasks and minimize the sum of the average travel cost per task over all the workers
- Offline Latency-oriented Task Completion problem [129]: The objective is to minimize the latency or the number of workers required for performing a set of tasks with high quality.
- Weighted Spatial Matching [121]: The objective is to find a fair assignment where there are no unstable pairs.

Table 2. Workers with preferred tasks and associated rewards

Worker ID	Preferred Spatial Task	Reward
W_1	t_1	5
	t_2	3
W_2	t_2	6
	t_3	2
W_3	t_5	8

Table 3. Assignment with Voronoi Diagrams

Worker ID	Assigned Spatial Tasks
W_1	t_1
W_2	t_2, t_3, t_4
W_3	t_5

- Minimum-cost Maximum Task Assignment Problem [61, 107]: The objective is to maximize the number of assigned tasks while minimizing the total travel cost.
- Maximum Quality Task Assignment Problem [16]: The objective is to maximize the overall quality score of assignments under travelling budget constraints across multiple time instances.

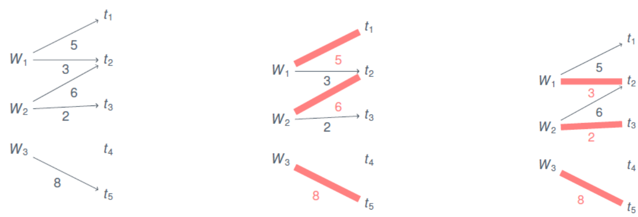
The ways to address this maximum task assignment optimization problem, by considering the constraints, will be discussed in the next sections. To understand the offline variant of the task matching problem, let us consider the following example.

Assignment Problem Example: Fig. 5(a) shows three workers with five spatial tasks in a *2D* region. The workers $\{W_1, W_2, W_3\}$ are required to collect pictures from the locations of the spatial tasks $\{t_1, t_2, t_3, t_4, t_5\}$. The spatial tasks were provided to the SC-server by the requester. Table 2 provides the information of worker task preferences and rewards associated with tasks. Our goal is to assign workers to these tasks based on the conditions mentioned by the requester. Kindly note that the example problem will be used throughout this survey to understand the different algorithms mentioned in the SC literature.

Fig. 5. (a). Base Problem: $\{W_i\}$ represents workers and $\{t_i\}$ represents spatial tasks. (b). Task Matching: Voronoi cells of the workers' locations.

Devoid of any constraints, the above mentioned task matching problem can be solved by assigning the tasks near the workers [60]. To solve this problem, the SC-server computes a *Voronoi diagram* based on the locations sent by the workers at a particular time instance. Subsequently, the SC-server assigns the spatial tasks close to the workers based on their *Voronoi cells*. Applying this solution to our SC assignment problem, a *Voronoi diagram* with the three workers in the *2D* region is generated(see Fig. 5(b)) and the assignment is mentioned in Table 3.

However with the constraints mentioned in Table 2, the task matching problem with the goal of maximizing the rewards received by workers, is solved by reducing it to a *Maximum Weighted Bipartite Matching* (MWBM) problem [94]. To reduce the SC task matching problem to the MWBM problem, a bipartite graph ($G = (V, E)$) is created with vertices $V = \mathcal{W} \cup \mathcal{T}$, where each worker W_j maps to a vertex in set $\mathcal{W}$ and each task t_k maps to a vertex in set $\mathcal{T}$ [107]. A vertex W_j in $\mathcal{W}$ is connected to a vertex t_k in $\mathcal{T}$ with an edge $e_{j,k} \in E$, if the task t_k is a preferred task of the worker W_j (see Fig. 6(a)). The weight of the edge $e_{j,k}$ is the reward associated with the preferred



(a) Bipartite Graph (b) Offline (c) Online

Fig. 6. Offline and Online Scenario: Maximum Weighted Bipartite Matching with total reward of 19 and 13 respectively

task. By solving the MWBM problem with the assumption that one worker can only perform one task, the workers are assigned to the tasks that give them the highest reward [107]. The worker W_1 has edges with both t_1 and t_2, however t_1 is assigned as it offers better reward than t_1. Similarly, W_2 has been assigned to t_2 and W_3 has been assigned to t_5 (see Fig. 6(b)). As none of the workers preferred t_4, it is not assigned to anyone. A total reward of 19 is achieved in the offline scenario.

4.3 Online Scenario: Dynamic Arrival of Tasks and workers

In the above example, all the workers and tasks are known to the SC-server, however, in a real-world scenario, the solicitation of tasks and availability of workers are dynamic in nature. The number of tasks cannot be constant as the requesters can provide new tasks to the SC-server at every instance of time and the existing tasks can expire once their deadline is reached. Similarly, regarding the number of available workers, they can increase or decrease at every instance of time. Therefore, the SC-server does not have the *global knowledge* required for assigning the spatial tasks to the workers in a globally optimal way. This input model scenario is referred as the **online scenario** [58]. In other words, the SC-server has no information regarding the arrival orders of the workers and tasks to the system. Owing to this constraint, a *locally optimal* solution can be found for the spatial task assignment at every time instance. The locally optimal solution at every time instance is then combined to provide the solution of the online task matching problem.

A locally optimal solution is akin to the offline scenario, as the available workers and tasks are known to the system at the current time instance. Therefore, the above mentioned offline methods can be employed to solve the optimization problem at every time instance. However, in the case of online scenario, only a partial bipartite graph can be constructed as the arrival order of tasks and workers is unknown. Therefore, the locally optimal assignment will be performed on the partial bipartite graph at each time instance, resulting in sub-optimal results. Considering the previous example in Section 4.2, let us consider the order of arrival of tasks and workers as $<W_1, t_2, W_2, t_4, t_1, W_3, t_3, t_5>$. When t_2 arrives, the available worker W_1 will be assigned to it, even though it offers a lesser reward when compared to t_1, that arrives at a later time instance. Similarly W_2 is assigned to t_3 and W_3 to t_5 (see Fig. 6(c)). The total reward is reduced to 13, whereas offline scenario achieved 19. Therefore, the effectiveness of the online task assignment is dependent on the arrival order of the tasks and workers.

To address the online task assignment problem efficiently, many solutions are proposed like Least Location Entropy Priority (LLEP) [61, 107], multi-armed bandit formalization of the online task assignment problem [47], Two-phase based Global Online Allocation (TGOA) algorithm, Global Online Micro-task Allocation (GOMA) [111], Online Fixed/Dynamic-based Maximum Task

Coverage Algorithm [113], Hierarchically Separated Tree based randomized online algorithms (HST-Greedy) [110], and Prediction-oriented Online Task Assignment in Real-time Spatial Data(POLAR) algorithm [112].

To tackle the randomness of the online task assignment problem, the historical information regarding the workers' movement behaviour, tasks' location information, and workers' task acceptance behaviour is utilized. The Least Location Entropy Priority strategy [61, 107, 113] improves on the greedy strategy of finding locally optimal solutions at each time instance, by exploiting the workers' historical information shared to the SC-server. Location Entropy [23] estimates the diversity of unique visitors to a location, in this case, location is assumed to be a grid cell. Higher location entropy is synonymous with more workers visiting the location with an even distribution of visits among the workers. Therefore, tasks lying in areas with lower location entropy have lesser chances to be completed as fewer workers visit those locations. Higher priority is given to the tasks present in smaller location entropy areas to maximize the assigned tasks, by reducing it to a *minimum-cost MWBM* problem, i.e., finding the MWBM of minimal total cost/entropy. The minimum-cost MWBM problem is solved in two steps: finding the maximum matching using the Hungarian algorithm and minimizing the cost of the matching by applying integer programming technique like the branch and bound. Consider the example in Fig. 6(c), it represents the output of the maximal matching of the MWBM problem. In the minimum-cost MWBM problem, the cost of the edges is set as the entropy of the workers at the locations of the tasks. According to [107], there will be a 35 % improvement in the performance of LLEP when compared to the greedy approach. Therefore, the LLEP approach would result in a total reward of 17.

Furthermore, the tasks assigned by the SC-server could be rejected or accepted by the worker, adding to the complexity of online task assignment problem. Therefore, a task assignment will be considered as successful only if the worker accepts to perform it. In [47], the authors propose an IMIRT (Individualized Models for Intelligent Routing of Tasks) framework for online task assignment, which focuses on modelling the task acceptance behaviour of the workers from the past data. Subsequently, the task acceptance behavior models are utilized for optimizing the number of successful assignments. This framework assumes that the workers are known to the SC-server, and only the tasks appear dynamically.

The framework profiles the task acceptance behaviour of the workers and utilizes that information to improve/maximize the assignment success rate, i.e., the ratio of the number of tasks workers accept to perform against the total number of assigned tasks by the SC-server. The SC-server checks for the workers and would assign the task to the ones with a higher probability of accepting the task. The online assignment problem is formalized as a multi-armed bandit [91] model, where each worker W_j is considered as an arm, assignment $a_{i,j}$ of the task t_i is equivalent to playing an arm and the resulting reward $y_{i,j}$ is the probability of success. For a newly arriving task, selecting a worker with the highest probability of success $y_{i,j}$ would result in maximization of the assignment success rate. The type of task acceptance behavior model has a huge impact on the online spatial task assignment. For instance, if each worker has a fixed behavior of task acceptance, i.e., $p_{i,j} = p_{i',j}$ for any tasks i and i', it could be modeled as a Binomial process with parameter p_j. To improve the efficiency of task assignment, a new strategy *SpatialUCB* is proposed by integrating task acceptance behavior model with spatial contextual information like travel distance from task t_i to worker W_j location $d_{i,j}$ and type of task e_i.

The IMIRT framework [47] only takes into account the dynamic nature of tasks, and assumes the workers to be static. In [111], the authors proposed algorithms based on the online model where both tasks and workers can appear dynamically. A Two-phase-based Global Online Allocation (TGOA) algorithm is proposed that divides the set of workers and tasks into two equal groups based on their arrival orders. The input set of workers and tasks is estimated through the historical records for a given time interval, thereby eliminating the uncertainty of arrival order. The greedy strategy would be applied to the first half of the input set, where an arriving task or worker would be paired with its complementary that has highest utility value. For the second half of the data set, an optimal strategy is adopted to find the optimal global match to the arriving worker/task. This approach provides competitive ratio guarantee of 1/4 under the online random order model. Considering the previous example in Section 4.2, let us consider the order of arrival of tasks and workers as <W_1, t_1, W_2, t_4, t_2, W_3, t_3, t_5>. THE TGOA algorithm splits the input set into 2 sets of 4 according to their arrival orders, i.e. <W_1, t_1, W_2, t_4> and <t_2, W_3, t_3, t_5>. On the first half of the vertices, TGOA performs greedy assignment, therefore W_1 is assigned to t_1 and W_2 is assigned to t_4. For the second half of the vertices, TGOA performs a global optimal match, therefore, W_3 is assigned to t_5. The total reward achieved through this method is 15.

The above approaches assume that the worker would either be assigned to a task immediately after being available to the SC-server or waits for a task at the same reported location until the specified deadline. This assumption is impractical as the worker would, most likely, not stay idle at the same location waiting for a task to be assigned. The worker tends to roam around in the hope of finding a task to perform. Moreover, if the worker stays at the same location waiting for the task, then she might miss out on many potentially matching tasks that appear in a different neighborhood.

Instead of making the worker wait, it would be beneficial to guide the worker to a neighborhood where new potential matching tasks might appear. [112] proposes a two-step framework, the *Flexible Two-sided Online Task Assignment* (FTOA) model. The framework allows workers to either stay at the same location waiting for a task or to be guided to a different location if the worker is not assigned a task on being available to SC-server. This approach utilizes the historical data for predicting the number of tasks and workers in a specific spatiotemporal range. The spatial and temporal dimensions are partitioned into *grids* and *time slots*, respectively. Subsequently, an offline guide is created by performing offline matching of the predicted tasks and workers according to the spatiotemporal divisions (grids and time slots) and deadline constraints.

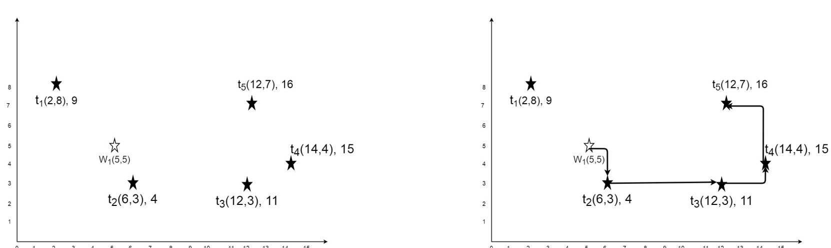
The offline guide contains the potential task-worker assigned pairs with individual nodes representing tasks/workers in a particular grid at a specific slot of time (the grid-time slot pair is referred to as object type). This offline guide would be used in *Prediction-based Online task Assignment in Real-time Spatial Data* (POLAR) algorithm. According to the algorithm, a newly arrived task or worker would be matched with the existing nodes of the same type (same time slot and grid) in the offline guide. The matched node would be occupied by the newly arrived task/worker (object). Each node of the offline guide can only be occupied by one task/worker of the same type. If there exists a real worker/task corresponding to the occupied node in the offline guide, the nodes are assigned to each other in the online model. If no actual tasks are corresponding to the occupied nodes in the offline guide, then the algorithm would suggest the worker to move to a grid where the potential matching task would appear at a future time. However, there might be some unpredicted tasks/workers appearing to the SC-server. The POLAR algorithm ignores such unpredicted tasks/workers as they cannot be matched with the nodes of the offline guide. An optimized POLAR algorithm (POLAR-OP) is proposed to handle the unpredicted tasks/workers. The POLAR-OP allows the nodes of the offline guide to be associated with more than one task/worker, i.e., each node of the offline guide can be reused by multiple objects.

The competitive ratios of POLAR and POLAR-OP are 0.4 and 0.47, respectively, under the independently and identically distribution (i.i.d) model [36]. Both POLAR and POLAR-OP, processes each newly arrived task/worker by looking up the offline guide in $O(1)$ time. However, the success of this approach is dependent on the accuracy of the prediction model used in the initial stage. Furthermore, this approach of moving the workers in advance is not beneficial in the cases where there are more workers than the tasks. In such cases, simple greedy approaches outperform the POLAR algorithm.

5 TASK SCHEDULING PROBLEM

5.1 Introduction

In the previous Section 4, task matching problems focus on assigning tasks to workers. However, to complete a task, the worker has to travel to the physical location of the assigned task. Given a set of assigned/selected tasks, travelling to every task before their respective deadlines might not be possible as there might exist tasks with overlapping deadlines, long travel times between task locations, etc. Therefore, a plan is needed to complete as many assigned/selected tasks as possible, to maximize the reward collected, leading to the optimization problem of creating a schedule/task sequence that maximizes the number of tasks performed by the worker. The spatial tasks sequence is generated taking into consideration both the travel cost and expiration time of the tasks. This optimization problem is referred as *Maximum Task Scheduling* (MTS) problem [29].



(a) (b)

Fig. 7. Maximum Task Scheduling Problem in WST scenario: (a). Worker W_1 with the tasks $\{t_1, t_2, t_3, t_4, t_5\}$ along with their locations and expiration time, and (b). Maximum Valid Task Sequence of the MTS problem

5.2 Offline Input Model: One worker- Multiple Tasks Scenario

The maximum task scheduling problem [29] addresses the offline variant of task scheduling problems, where all the input parameters are known to the SC-server beforehand. The SC-server knows the worker and the set of tasks assigned/self-selected by her. To elaborate, let us consider the SC task scheduling scenario with a set of four tasks $\{t_1, t_2, t_3, t_4\}$ selected by the worker W_1 (see Fig. 7(a)). Each task has an expiration time d_i and the worker needs to travel to that task location before the expiration time. In this example, let us assume that the cost of travelling between two adjacent grid cells is one time unit. The worker initial location is (5,5) at time *0*. The task t_2 is located at (6,3) that is scheduled to expire after 4 time units. The travel cost between the two tasks' locations is 9.

The worker has to choose the relevant subsets or the sequence of tasks to maximize the number of accomplished tasks considering the travel costs as well as deadlines of the task sequence. A sequence of tasks where all of its tasks can be completed is termed as *valid task sequence* and the one corresponding to the maximum number of tasks is termed as *Maximum Valid Task Sequence*. *Maximum Task Scheduling* (MTS) problem is to find this *Maximum Valid Task Sequence*. MTS problem varies from the general job scheduling problems [82] with respect to the time required for performing the task. In job scheduling problems, the time required for each job is known in advance, whereas in the MTS problem the time required is dependent upon the travel time between the tasks, which in turn depends on the orders in which tasks are performed.

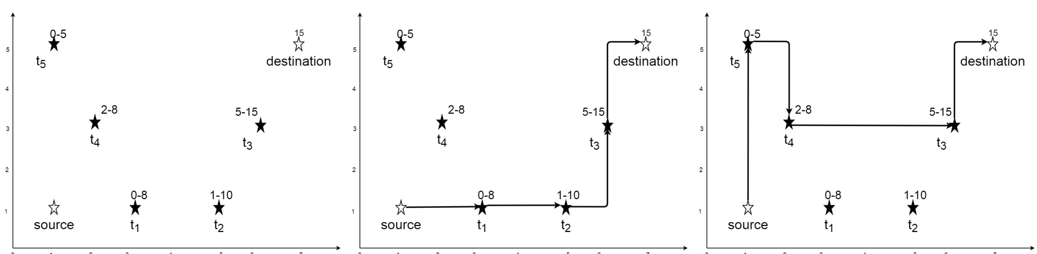
For example in Fig. 7(b), if the worker starts with the task t_1, then the travel cost from the worker's current location to the task t_1 is 6 time units and the deadline to perform t_1 is 9 time units. As the worker reaches t_1 at time *6* i.e., before the deadline of t_1, therefore the task t_1 can be performed by the worker W_1. At time *6*, the worker cannot perform the task t_2 since the deadline for it (4 time units) has passed. Similarly, the tasks t_3, t_4 and t_5 cannot be performed, as the travel cost exceeds the deadline of those tasks, i.e., travel cost + current time for all these tasks exceed the maximum deadline period of the tasks, i.e., deadline of t_5 (16). Similarly, if the worker starts from the task t_2, then she could be able to complete the tasks t_3, t_4 *and* t_5 as well (see Fig. 7(b)). Therefore, choosing the sequence of performing the tasks plays a major role in determining the number of tasks performed by the worker. A longer sequence of the tasks subset would result in higher number of tasks performed.

Maximum Task Scheduling (MTS) problem is proved as NP-hard in [29] by reducing it to a specialized version of Traveling Salesman Problem. For a small set of tasks, the MTS problem can be solved by finding the *Maximum Valid Task Sequence* using the brute-force approach. In this method, they check for all possible combinations of the valid task sequences and check for the sequence with the maximum number of completed tasks. However, this method is computationally expensive as computing all task sequences for n tasks will be $O(n!)$. Two strategies (exact algorithms) were proposed based on dynamic programming and branch-and-bound strategy, to address this issue.

Exact Algorithms: The dynamic programming approach investigates the sets of tasks ignoring the order of task sequence. The search space is generated by expanding the sets of tasks iteratively in ascending order of set size from 2 to n. Therefore, $2^n - 1$ subsets of tasks need to be investigated to find the *Maximum Valid Task Sequence*. This approach is optimized further by removing the invalid subsets of tasks from examination. An invalid set is defined as a task set without any valid task sequence in its combinations. In the branch-and-bound approach, the search space is represented as a tree, where depth-first search along with pruning unpromising branches is conducted recursively, till a feasible solution with *Maximum Valid Task Sequence* is found. The branch-and-bound approach is more efficient than the dynamic programming approach regarding space requirements.

In the dynamic programming approach, there are at most $O(n \cdot 2^n)$ subproblems, and each one can be solve in linear time. Therefore, the time complexity of dynamic programming is $O(n^2 \cdot 2^n)$, which is faster than the brute-force approach's $O(n!)$ running time. On the other hand, the time complexity of the branch-and-bound approach is proportional to the search tree size. If the branches are pruned, and all the unnecessary nodes are removed then the time complexity of the branch-and-bound would be better than dynamic programming. It should be noted that if all the branches of the tree needs to be searched without pruning branches, then the worst case time complexity of branch-and-bound would be $O(n!)$. However, in practice it is a rarity for a tree to be searched without pruning the branches, therefore branch-and-bound approach would be better than the brute-force approach.

Approximation Algorithms: Both the branch-and-bound and dynamic programming approaches result in an exponential time complexity, which is not suitable for real-world applications. Therefore, approximation algorithms are proposed in [29] based on different heuristics like *Least Expiration Time Heuristic*, *Nearest Neighbor Heuristic* and *Most Promising Heuristic*. With the Least Expiration Time Heuristic, task sequences are formed greedily by selecting tasks with least expiration time. In the case of Nearest Neighbor Heuristic, the nearest available task to the last task in the sequence is added. Finally, the Most Promising Heuristic helps in choosing the most promising branch in each iteration instead of searching the entire tree. The approximation algorithms output with less accuracy compared with exact algorithms but quicker response. It was noticed that in real-world applications, it is sufficient to report a small number of tasks assigned to a worker, while the remaining solution is computed offline by the server. This leads to the proposal of progressive algorithms [29], where approximation algorithms are used to find out a small number of tasks for a given worker. The optimal task sequence for the remaining tasks is found out by the exact algorithms like branch-and-bound. Progressive algorithms improve the response time when compared to exact algorithms and accuracy when compared to approximation algorithms. However, there are chances that worker's potential tasks may be arrogated by other workers, when the worker is on the way to perform the initial tasks. Moreover, workers may prefer to see the entire task sequence before starting work.



(a) (b) (c)

Fig. 8. Online Route Recommendation Problem example with source, destination and arrival time at destination=15 in WST scenario: (a). Worker W_1 with the tasks $\{t_1, t_2, t_3, t_4, t_5\}$ along with their locations, release time and expiration time. (b). Result Route of the OnlineRR problem with Nearest Neighbor Heuristic. (c). Result Route of the OnlineRR problem with Earliest Deadline Heursitic

5.3 Online Input Model: One Worker- Multiple Tasks Scenario

One of the limitations of the MTS problem is that it calculates the maximum valid task sequence for a particular snapshot of time. The outcome task sequence does not update according to the arrival of new tasks that lie in the spatial region of the worker, which are requested to the SC system by the requesters. Furthermore, according to [2] workers often take tasks that are present in their route, for example, daily commute from home to work, resulting in extra constraints like destination and arrival time at the destination. Therefore, considering both these requirements would further improve the outcome of the task scheduling problem. Taking both these requirements into consideration, [73] introduced an *Online Route Recommendation Problem* (OnlineRR) for workers who choose their tasks.

For example, consider Fig. 8(a) which shows a set of tasks $\{t_1, t_2, t_3, t_4, t_5\}$ along with their locations, release times, and deadlines. The figure also shows the source and destination points of the worker W_1 and the time taken to arrive at the destination (15). The OnlineRR finds the longest sequence of tasks which can be performed considering that the worker should be present at the destination at time 15. To solve this problem, [73] presents two approaches: greedy approach and re-routing through a complete search at a time snapshot.

The greedy approach was chosen with the aim of reducing response times since the global view of the problem is not possible due to the continuous task and workers' location updates. In the greedy approach, the next task is chosen according to heuristics, such as the nearest task to the worker's current location, tasks with earliest deadlines and, maximizing the search space of feasible tasks. Furthermore, for choosing the next task it should satisfy the condition of reaching the task's location before the task's deadline.

For instance, consider Fig. 8(b), it depicts the result route outcome of the greedy approach with the nearest neighbor heuristic. The worker at time *0* is at the source location, and she searches for the closest task in her vicinity. t_1 is selected as it is the nearest with 2 units distance and takes 2 time units to travel to task's location. Now the current location of the worker (at time *2*) is updated to the t_1 location. Worker again begins her search for the nearest neighbor whose travel distance is less than the deadline minus the current time, in this case, it's t_2 that is 2 units away from t_1. The current location will be updated to t_2 at time 4. Similarly, t_3 will be selected and the current location will be updated to it at time 7 and the route to (source, t_1, t_2, t_3). Now at time 7, there are no feasible tasks available to perform as the worker cannot reach the tasks' location the deadlines. Therefore, the worker heads to the destination which is 3 time units away. Finally at time 10, the destination of the worker will be reached and the result route will be (source, t_1, t_2, t_3, destination).

Similarly, in Fig. 8(c), the result route of the greedy approach with earliest deadline heuristic is depicted. The difference from the nearest neighbor is that the preference or high score will be given to tasks that have earlier deadlines than the remaining available deadlines. For example, at time 0, the earliest deadline for a task is t_5 with the deadline at 5 time units, consequently t_5 is selected. In similar fashion, maximum candidate space heuristic is used to calculate the score of tasks based on the distance.

The second approach to solve the OnlineRR problem is the re-routing method, where the optimal solution is calculated at the current snapshots recursively as long as there exists a new feasible task p. For instance, consider the example in Fig. 8(a), at time 0, worker has two feasible tasks (t_1, t_5) and re-route algorithm computes the task sequence R_0 (source, t_1, t_5, destination). According to the task sequence, the worker moves to t_1 at time 2 and new feasible tasks are found $\{t_2, t_4\}$. Subsequently, the task sequence is re-computed to R_1 (t_1, t_4, destination). Following the re-computed task sequence, the worker moves to t_4 at time 5 and found one new feasible task t_3 and the new route R_2 (t_4,t_3, destination). Following the new route, the worker moves to t_3 at time 9, where there are no more feasible tasks. Therefore, the worker has travelled the following route (source, t_1, t_5, t_4, t_3, destination). It covers more tasks than the other heuristics mentioned earlier.

The previous scheduling problem deals with a single worker along with multiple point tasks. The problem gets complicated while scheduling for delivery tasks that have both the source and the destination. The problem of scheduling delivery tasks for a single worker, The *Online Delivery Route Recommendation Problem* is discussed in [100]. The optimization problem objective is to maximize the worker's income. Following a prediction-based approach, an algorithm is proposed considering three factors, namely; the distance between worker's location and origin of the feasible delivery task, the distance between origin and destination of the feasible delivery task, and the future demand originating from the destination of each delivery task. This algorithm has a shortcoming if the worker has a personal destination that she should reach before a deadline. With the prediction-based approach, it is possible to be far away from the destination and having to spend much time to return to the destination, consequently losing on the potential tasks and rewards. To address this shortcoming, the prediction-based algorithm has been extended [100], to consider the impact of distance between delivery task's destination and worker's destination. The current approach can plan for a group of delivery tasks for a single worker. New methods needs to be proposed for tackling the scenario of multiple workers with multiple delivery tasks.

5.4 Combination with Task Matching Problems: Multiple Workers-Multiple Tasks Scenario

Table 4 summarizes the different task matching and task scheduling problems in the SC literature. Task scheduling is necessary to assist the worker to complete as many assigned tasks as possible. However, there are some drawbacks of the above-discussed task scheduling problems. They do not address the issue of re-matching the tasks that cannot be performed by the worker with other available workers and subsequent re-scheduling for the re-matched tasks. Consequently, inefficient scheduling results are produced in multiple workers scenario. One solution to address these shortcomings is to combine task matching and scheduling problems and carry them out iteratively. This combination approach would improve the task acceptance and completion rates due to the enhanced awareness of SC-server. The SC-server can determine whether a matched task can be completed by the worker through task scheduling algorithms.

As observed in Table 4, there are examples of SC optimization problems that combine both task matching and task scheduling. Furthermore, the previously discussed task scheduling problems focus on a single worker and multiple tasks scenario. In [30], a combination of task matching and scheduling problem is proposed to minimize the travel cost of multiple workers and maximize the number of completed tasks. The authors propose two solutions to address this problem: *Global Assignment and Local Scheduling* (GALS) and *Local Assignment and Local Scheduling* (LALS). GALS approach performs task matching and scheduling sequentially and utilizes the output of task scheduling to refine the task assignments. To elaborate, GALS performs an initial task assignment resulting in a set of tasks assigned to each worker, subsequently assigned tasks are scheduled for each worker and finally, the task assignment is refined with the unscheduled tasks from the previous step and available workers. GALS approach performs the scheduling and updating task assignment steps iteratively until the termination condition is reached. The iteration condition is satisfied when there are no potential worker-task pairs available in the input set for updating task assignments. The initial global task assignment and assignment update phase are computationally expensive than the local scheduling phase, due to the size of search space for performing the assignments.

Bisection-based LALS approach is proposed to improve the scalability of GALS approach. Initially, this approach recursively divides the input dataset of workers and tasks in a top-down approach, until the size of the partition is less than a predefined threshold. Then, the resulting partitions are merged in a bottom-up approach according to a predefined threshold. If the combined workload of the sibling partitions is less than the threshold, then the sibling partitions are merged to create a new partition. However, if the combined workload of the two sibling partitions is more than the threshold, then a local assignment and scheduling is performed on the individual sibling partitions. The remaining workers and tasks from the individual partitions are then merged to create a new partition. The merging is performed iteratively until the root node is reached in the partition tree. The intuition behind merging the remaining workers and tasks of the sibling partitions is to utilize the spatial properties of the binary tree for improving the quality of assignments and scheduling. Similarly, in [12], a coordinated task assignment approach, Trajectory-Aware Coordinated Urban Crowd-Sourcing (TRACCS), is proposed to assign a sequence of tasks to worker considering their movement patterns.

In the above approaches, task assignment and subsequent scheduling is performed in batches. However, in real-time online assignments, it would be necessary to assign and schedule tasks as soon as they enter the system. Consequently, the SC-server is performing schedule operations for multiple workers for every new task, which is difficult to support and thus results in scalability issues [5]. To tackle this problem, [5] proposes a novel auction-based framework Auction-SC, where the workload is split between the workers and the SC-server. Workers bid on the newly arriving tasks, and the highest bidder would get the task assigned to them based on the auction framework proposed in [4]. The workers would submit their schedules to the SC-server to assess on the feasibility of the new task to their valid schedule and for computing their optimal bids. Hence, this approach tries to benefit both the worker by maximizing her profits and the SC-server by maximizing the number of completed tasks. This approach does not consider the time required for performing a task, and does not support complex tasks. Moreover, the case where the worker fails to adhere to the task sequence is not considered. Similarly, [135] proposes a solution to the offline destination aware task assignment problem, that utilizes task dependencies among workers, for dividing them into independent clusters and deadline constraints of workers' destination and tasks' expiration time. This approach would result in an optimal assignment without the need of re-assignment and re-scheduling. However, this approach does not consider the time required for performing a task which would have a more significant impact during scheduling. Furthermore, it does not consider the dynamic arrival of tasks and workers.

6 SPATIAL CONSTRAINTS

6.1 Overview

Spatial Constraints refer to the different spatial preferences of the worker or the task. The SC-server assigns or publishes tasks to the workers, while satisfying specified constraints. In the case of Worker Selected Tasks (WST) publishing mode, it is hard to enforce the spatial constraints without the knowledge of worker's location. Some of the applications like Gigwalk[7], TaskRabbit[8] enquire the workers location or pin code to serve the tasks in their proximity. After the selection of tasks in the WST mode, spatial constraints can be enforced while scheduling a plan for completing the tasks. For example, in [73] there are spatial constraints regarding route planning starting from the source and reaching the destination at a particular time. The SC-server can include as many tasks as possible in the plan, provided these conditions were met.

On the contrary, in the case of Server Assigned Tasks (SAT) publishing mode, spatial constraints can be strictly enforced. Some of the types of spatial constraints are the spatial region, maximum travel distance, and direction of worker's commute. The spatial constraints are usually dependent on the current location of the worker, for example, preferring tasks that are within 500 metres from the current location of the worker. In the case of the offline scenario, the worker's location remains static or known to the SC-server, thus the spatial constraints are not expensive to construct. However, in the case of online scenario, the worker's location is received by the SC-server in a streaming fashion, therefore a change in worker's location triggers a re-construction of spatial constraints of the worker.

6.2 Spatial Region

A worker can define his preferred spatial region as a minimum bounding rectangle [61, 62, 107] or as a bounding circle with her location [111]. It is assumed that the worker is willing to travel to any of the tasks that lie within the defined spatial ranges. Similarly, requester can specify spatial constraints on the task, like in the case of region tasks [103, 113] where the worker could be located anywhere in a given circular range.

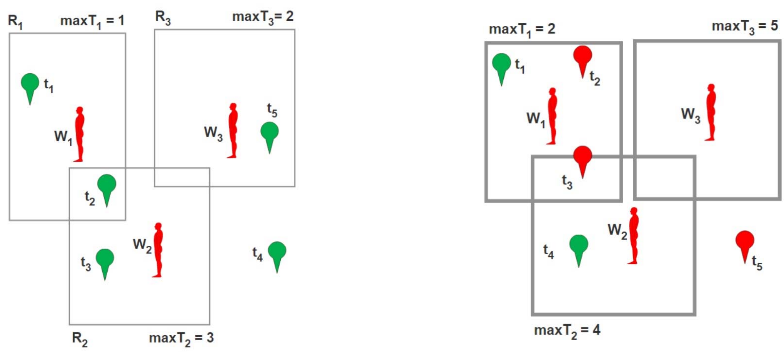
Example. To understand the effect of spatial constraints on a task assignment, let us consider the basic assignment problem of Section 4.2 and enforce the spatial region constraint of the workers. Fig. 9(a) showcases an example scenario representing the spatial tasks and the workers along with their preferred spatial regions and the maximum number of tasks that the worker can perform. As discussed in previous sections regarding the task assignments, there exists a many-to-many relationship between tasks and workers, for instance, a task can be assigned to multiple workers and vice versa. However, the relationship is subject to satisfying the conditions: the task should lie in the preferred spatial region of the worker, and the threshold of the maximum number of tasks performed by worker is not violated.

One possible assignment would be between the spatial task t_5 and worker W_3 as the task lies in the region of R_3 and the $maxT_3$ value is 2. Similarly, spatial tasks t_1 *and* t_2 lies in the region R_1 of

[7]http://www.gigwalk.com/
[8]https://www.taskrabbit.com/

Table 4. Summary of the Surveyed Task Assignment and Scheduling Problems: The acronyms used according to category are: **Constraints**: SpC -Spatial, TC -Temporal, BC -Budget, QC- Quality. **Dynamic nature of Inputs**: TA, WA -Task and Worker Arrival. **Heuristics**: Gr-Greedy, TrC-Travel Cost, LE-Location Entropy, NN-Nearest Neighbor. **Privacy**: PP- Privacy Protection

Problem Type	Problem Name	Support for Constraints				PP	Dynamic Arrival		Reduced Problem	Algorithm Heuristic	Limitations
		SpC	TC	QC	BC		TA	WA			
Task Matching	MTA [61]	X	X						Max-Flow MinCost-MaxFlow	Gr LE TC	a. Does not work well in a dynamic setting, esp. greedy strategy. b. Workers are assumed to accept tasks that are assigned to them. c. Trusts the worker to perform the task correctly. d. Does not consider the worker movement patterns
	MSA [107]	X	X	X					MWBM MinCost-MWBM	Gr LE TC	
	MTMCA [27]	X	X	X	X				MinCost-MaxFlow	Reward	
	RDB-SC [17]	X	X	X					Number Partition	Gr	a. Worker is assumed to be constantly moving in a specific direction with a certain velocity. b. No Option to limit the distance a W can travel, resulting in assignment of far away tasks.
	MRA [133]		X	X					SCP [6]	Gr	a. Does not take into account the movement of workers and the dynamic arrival of tasks
	PAPA [60]	X				X			Set Cover	Gr	a. k-anonymity based approach has limitations like it is prone to homogeneity attack, or the privacy relies on the value of k.
	DPTA [105]	X				X				TC	a. Does not take into account the movement of workers b. Produces sub-optimal assignment due to fake workers in the geocast region.
	MS-SC [15]	X	X	X					Set Cover	Gr g-D&C Adaptive	a. Assumes the cost of performing a task proportional to travel cost, and Euclidean distance function is used to calculate travel cost.
	OSTA [47]						X		MAB	Spatial Context	a. Does not account for the dynamic nature of workers b. Only one worker can be assigned to the task
	Fixed Budget MTC [103, 113]	X	X		X				MCG	Gr:Offline	a. Limited to hyper local tasks that do not require the workers to travel. b. Worker cannot try to maximize his reward by performing sequence of tasks. c. The performance is hindered due to the dynamic arrival of workers, as the budget is allocated per time periods d.Offline algorithms, though not practical, helps in tackling the randomness of the real-world online problem.
							X		MCP	Gr Temporal LE	
	Dynamic Budget MTC [103, 113]	X	X		X				MCP	Gr:Offline	
							X		MAB	Gr Temporal LE	
	FTOA [112]	X	X				X	X	Offline Guide: Max-Flow	TrC	a. Dependent on the accuracy of the initial prediction model b. Not beneficial when the workers are more in number than the tasks.
	GOMA [111]	X	X	X	X		X	X	OMWBM	Gr	a. Focuses on micro-tasks that are simple and trivial. b. Focuses on maximizing the benefit of SC-server and not the worker.
Task Scheduling	MTS [29]	X	X		X				TSP	Temporal NN MPH	a.Not applicable in the case of dynamic arrival of tasks b. The set of tasks is preselected for the worker.
	OnlineRR [73]	X	X				X		OPTW	Temporal NN MCS	a. Set of tasks arrive dynamically, however schedules plan for only one worker. b. Does not take into account some constraints like quality
Task Matching & Scheduling	MTSMW [30]	X	X						Matching: MTA Scheduling: Routing	Same as MTA TC	a.Assignment module has similar limitations as the MTA problem b. Assumes new task insertion does not effect task order.
	TRACCS[12]	X	X						Orienteering	Gr ILS	a. Partial order is maintained among the nodes visited. b. Preference to individual benefits than globally optimal solution.
	OnlineTASC [5]	X	X		X		X	X	Min-Len-Ham	Random Ranking NN Most Free Time Best Insertion Best Dist.	a. Does not consider the required time for completion of task b. Does not support complex tasks
	DSTA [135]	X	X							Gr	a. Dynamic arrival of tasks and workers is not considered b. Does not consider the required time for completion of task



(a) (b)

Fig. 9. (a).SC in SAT mode with workers' constraints: Spatial Regions $\{R_\text{i}\}$ and the maximum number of tasks $\{maxT_\text{i}\}$ (b). An expertise matching example of workers set $\{W_i\}$ and spatial tasks $\{t_i\}$: Same colour(red) between workers and spatial tasks represent an expertise match.

the worker W_1. However, both the tasks cannot be assigned to W_1, due to the maximum number of tasks performed constraint ($maxT_1$ value is only 1). Therefore, either t_1 *or* t_2 will be assigned to W_1. Since, the spatial task t_2 lies also in the region of W_2, who has a higher value for $maxT$ (=3), the task t_2 is assigned to W_2 to maximize the number of assignments. On the other hand, as t_4 does not lie under any spatial region of the available workers, it remains unassigned. This assignment problem could be solved by the strategies proposed in [61] by finding out the locally optimal solution.

6.3 Maximum travel distance

A worker can set a maximum travelling distance she is willing to travel for performing the assigned tasks beyond her commute or movement trajectory [12]. The server should intuitively assign tasks to maximize the number of tasks performed and the overall rewards received without violating the maximum travelling distance threshold of individual workers.

6.4 Direction of worker's commute

The spatial constraint about the direction of worker's movement would help in assigning tasks to a worker without a significant deviation from her path [17]. For example, if the assigned task lies in the opposite direction of worker's movement in a directional road network, then the travel distance to the task would be longer due to the necessary direction change in worker's commute. Given the longer travel distance, the likelihood for a task to be rejected by the worker is high. Furthermore, assigning the tasks based on the direction of worker's commute might improve the diversity of spatial angles regarding the information queried by the task. For instance, if the task involves a taking a picture of a public monument, assigning the task to different workers who travel through the task location in different directions would result in a wide range of images taken from different angles, thus improving the diversity of spatial angles.

Table 5. Comparison of task assignment protocols with multiple constraints

Constraints	Spatial	Quality	Budget
Spatial	MTA [61]	MSA [107], RDB-SC [17], MCTA [62], GOMA [111], MTMCA [27]	f-MTC/d-MTC [103, 113], MTMCA [27], GOMA [111]
Quality	MSA [107], RDB-SC [17], MCTA [62], GOMA [111], MTMCA [27]	MRA [133]	MS-SC [15], MTMCA [27], GOMA [111], Crowdlet [88], MQA [16]
Budget	f-MTC/d-MTC [103, 113], MTMCA [27], GOMA [111]	MS-SC [15], MTMCA [27], GOMA [111], Crowdlet [88], MQA [16]	-NA-

7 QUALITY CONSTRAINTS

7.1 Overview

In this section, we focus on the different quality constraints that assure a certain level of quality in the task responses from the workers. In the WST publishing mode, the server can conduct some qualification tests related to the skills required for the task before allowing the worker to view the task like in the case of [92]. However, apart from this, the server cannot exert any control to assure quality. On the contrary, in the case of SAT mode, the server can exploit the workers' profiles and find reliable workers with necessary skills to deliver a better quality task responses [1]. Two aspects of a worker's profile could affect the quality of task's outcome, namely a worker's expertise and reputation/reliability. These aspects of a worker's profile would be utilized during assignment of tasks in the form of constraints. The constraints could be; a required expertise level in certain skills to perform a task, and a minimum reputation/reliability measure for a task to be assigned.

7.2 Worker's Expertise

Expertise indicates the capabilities/skills of the worker in performing a specific type of tasks. For example expertise in, taking a photograph, tutoring a high school subject, painting a house, etc. Expertise on a particular skill is usually measured as a degree, like how skilled a worker is in tutoring a subject. Worker can specify the skills or expertise she possesses to the SC-server through submitting documents proving their credentials or experience. Alternatively, the server has the capability to assess the skills of the worker based on the historical information of tasks performed and their respective rating from the requester. Various studies have proposed methods [15, 27, 107] in the SAT publishing mode that utilizes the expertise aspect to assign tasks to capable workers. In the works mentioned in earlier sections, the spatial task assignment has the assumption of uniform expertise among all the workers.

However, workers are more likely to have varying expertise, for example, some may be good in taking expert pictures, while others excel in describing things at a location. Similarly, the spatial tasks types may vary, for instance, taking a high-quality picture of a monument at a physical location. Therefore, if a worker has the expertise that matches the spatial task type, then the resulting quality would be better than if the skills do not match the task. [34] In [27], an expertise constraint is introduced to the task assignment problem where the expertise of the worker has to match the expertise required of the task. The worker is defined to have a particular expertise E_w with a degree D_w and the task with required degree D_t of expertise E_t. The assignment constraint for assigning a spatial expert task is defined as: $E_\mathrm{w} = E_\mathrm{t} \wedge D_\mathrm{w} \geq D_\mathrm{t}$ The expertise E_w or E_t refers to a specific skill, therefore the major limitation of this definition is that the worker's expertise is limited to only one skill. Similarly, the spatial task cannot demand expertise from the worker in multiple skills. The optimization problem has the goal of maximizing the number of assigned spatial expert tasks and minimizing the total cost reward to workers.

However, in a real-world scenario, a worker could possess expertise in multiple skills and a task would also require multiple types of skills. [107] addresses one of the limitations of [27] regarding the issue of multiple skills of worker. The expertise of the worker W is defined as a set of skills E and assigned required skill as a task type e attribute to the task t definition. A score is assigned to all potential worker-task pairs $< W, t >$, indicating the performance of the worker W based on her expertise. An expertise match $< W, t >$ is made when the worker W possesses the expected skill of the task t in the preferred spatial region R, i.e., $e \in E$, and a higher score is assigned to the expertise match. On the contrary, if there are only workers available without the necessary skills required for the task, then a *nonexpertise* match, provided the task satisfies the spatial constraints. A lower score is assigned to these *nonexpertise* matches. By maximizing the overall score of the assignments, the total number of expertise matches would be maximized.

To understand this scenario, let us extend the basic assignment problem by adding the spatial and expertise constraints (see Fig. 9(b)). Tasks and workers with the same expertise are represented in the same color. It can be noticed that the spatial task t_5 will not be assigned to any of the three workers. Furthermore, worker W_3 has no spatial tasks in its spatial region. Therefore none of the tasks will be assigned to her. It can be observed that there is an expertise match between W_1 with t_1, t_2 *and* t_3. Out of these, $< W_1, t_1 >$ is an non-expertise match and the remaining two matches $< W_1, t_2 >$ and $< W_1, t_3 >$ are expertise match. As observed, there are chances for an expertise match or non-expertise match. Each match will be given a score, and the aggregate score will be counted for each time instance. The problem is to maximize the aggregate score of the assignments made by the SC-server. This problem is called *Maximum Score Assignment Problem* [107], that can be reduced to the *Maximum Weighted Bipartite Matching Problem*. The method proposed in [107] does not address the case of the task requiring multiple skills for being assigned to a worker. [15] addresses this limitation by defining the skill set required for the task to be assigned.

The discussed works assume that the worker does not move dynamically in the geographic space. However, workers are more likely to move around. [16] addresses the maximum quality task assignment (MQA) problem that assigns moving workers to spatial tasks while satisfying the budget constraints of travel cost. The optimization target is to maximize the overall quality score, considering the current and future workers/tasks. An accurate prediction approach is proposed to predict the quality location distributions of workers and tasks.

7.3 Worker's Reputation/Reliability

A worker's reputation/reliability refers to the probability that a task is completed correctly by the worker. It reflects the quality of the task response that can be expected from the worker. All the previously mentioned works assume that the worker can be trusted and tasks can be performed correctly. Therefore, assigning a task to a single worker would suffice in getting the expected outcome. However, in often cases, the assumption does not hold true as the workers might not be skilled enough to perform the task correctly. Moreover, some of the workers may have the malicious intent to exploit the system. Consequently, the confidence probability of the tasks and the workers should be considered, resulting in more than one worker being assigned a task with an aggregate reputation score satisfying the required confidence probability [62].

In [62], a worker W is defined along with the reputation score r, i.e., probability to perform the task correctly. The task t is defined along with the confidence threshold α, which is the minimum reputation score required for an assignment. A correct match is defined when a task t is assigned to set of workers with an aggregate reputation score Σr_i greater than or equal to the task's confidence threshold α ($\Sigma r_i \geq \alpha$). However, this approach does not focus on the dynamic nature of arrival of tasks and workers. Furthermore, the movement of workers is not taken into account during task assignment.

[17] tries to address the shortcomings of [62], by integrating the movement of workers with the help of the direction of movement and velocity at which the worker is travelling. The knowledge of workers' movement is used to improve the diversity of task responses that could be received from different spatial viewpoints. For example, one worker can take a picture of a monument from the west-side and another from the east-side. They address the challenge of the online scenario by designing a grid index to enable efficient updates of workers/tasks. [88] considers the online scenario of workers recruitment process. This approach takes into account the worker ability model and proposes methods to improve the overall quality of the workers recruited.

The worker's reputation/reliability is estimated during the process of truth discovery through aggregation of crowdsourced data [45, 50, 85, 90]. In the subsequent sub-section we will discuss the truth discovery process and estimation of worker's reputation/reliability.

8 BUDGET CONSTRAINTS

8.1 Introduction

Spatial tasks involve an incentive for the worker who performs the task. The incentives could be intrinsic or extrinsic in nature [1]. The intrinsic incentives could be based on personal interest, altruism of the workers, and the extrinsic incentives involving monetary rewards [27, 48, 67], virtual credits [14, 66]. Associating financial rewards to tasks would encourage more workers to take up the task, and accelerates the completion of task [81]. With the increased worker participation, the overall quality of the outcome might improve, although not with guarantees. With monetary rewards involved, spatial tasks attract fraud workers trying to maximize their benefits by providing false information. In contrast, when a spatial task needs to be performed without any extrinsic rewards, then the chance of the task being carried out by a committed worker could be higher.

In often cases in the SC literature, the spatial tasks does not involve any monetary incentives. Such voluntary tasks may have less chance of attracting workers [2], compared with tasks with incentives. Especially for spatial tasks requiring workers with a particular skill set to perform the task [27, 107]. Usually, workers with the required expertise expects an associated reward for performing the task. These kinds of tasks are termed as *Spatial Expert Tasks*. Merely associating extrinsic incentives to the task would not be sufficient as there is a need to control the incentives for workers with respect to the quality of information provided and their associated costs, such as travel cost, smartphone battery consumption cost, data transmission cost, and manual efforts involved [64]. Moreover, there could be additional constraints involved such as budget constraints from the requester end and reliability requirements [35, 38, 39]. Therefore, appropriate incentive mechanisms are required for SC systems. The relationships between the spatial, quality and budget constraints are illustrated in Table 5.

8.2 Reward Models and Incentive Mechanisms

The reward models are defined based on the two types of incentive mechanisms: *platform-centric incentive mechanism* and *user-centric incentive mechanism* [126]. In the *platform-centric incentive mechanism*, the SC-server exercises control over the allocation of rewards to the workers. In the *user-centric incentive mechanism*, the workers exercise control over the payment, by denoting the price for which they are willing to perform the tasks. Based on the incentive mechanisms, the reward models are classified as:

- ***Platform-centric***:

- **Fixed reward for the task**: The rewards for the spatial tasks are fixed, and the worker, who performs the task, receives the same amount irrespective of her costs involved, and she does not have any say in it [25].
- **Dynamic reward for the task**: The rewards for the spatial tasks are dynamic, and the worker who performs the task receives the amount depending on the quality of service provided and the costs involved [78].
- **Fixed Budget for Time Periods**: The SC-server assigns a fixed budget for each time period, where the maximum number of workers should be selected in that time period with the budget [103, 113].
- **Dynamic Budget for Time Periods**: The SC-server assigns a fixed budget for the entire campaign, where the workers are selected wisely to allocate the total budget to the different time periods of the campaign [103, 113].

• ***User-centric***:
  - **Reward expected by the worker** The worker has the option to set the amount she would like to receive for performing tasks. This value could be dynamic, and that could be updated with the task enquiries sent by the worker to the SC-server [27, 124].

However, the support for the reward models mentioned above is dependent on the type of task publishing mode chosen. For instance, with the WST task publishing mode, *user-centric* reward models are not viable, as the SC-server lacks knowledge regarding the worker requirements. Although, workers can apply a filter to select tasks with a minimum reward from the set of available tasks. Moreover, the WST mode is not efficient for *platform-centric* reward models, as there is little control on screening the workers, apart from conducting qualification tests and post-processing the outcome delivered by the workers and assessing the quality to reward appropriately [2, 14, 92].

On the other hand, the SAT task publishing mode supports both the *platform-centric* and *user-centric* reward models. As the server chooses the worker that can work on the spatial tasks, the incentives could be managed according to the different constraints of the task and the worker. The truthfulness of the worker's costs could be analyzed, and the payments could be calculated [35], especially in the cases of complex tasks where a budget is provided for the entire complex task, that should be divided among the sub-tasks. The general assumption is that the worker would try to maximize her benefit by competing with her peers. Therefore in order to be truthful, the workers should specify their cost independent of the other workers costs [126].

The reward models in the SC literature are primarily of *platform-centric* nature, corresponding to the SAT task publishing mode. Consequently, the SC applications tend to focus on benefitting the SC-server. However, the optimization goals could be altered to benefit the SC-server or the workers. For example, the optimization problem proposed in [27] minimizes the rewards expended by the SC-server, thereby benefitting the SC-server. On the contrary, the optimization problem proposed in [111], focuses on benefitting the worker by maximizing the rewards received by the worker. Although, it assumes that the SC will benefit from the fees it earns on successfully assigned tasks. [48] proposes a pricing mechanism based on bargaining theory, to help both the SC-server and the workers to determine the rewards associated with the tasks. The reward of the tasks are dependent on the cost incurred by the workers to perform the task and the number of available workers. A higher number of available workers would result in lower rewards for tasks. Similarly, [19] utilizes the geo-social relationships to develop a diversity-driven and socially aware reward mechanism to improve the value of information by improving diversity among the recruited workers. [127] utilizes the fog platform [77] to identify the most valuable workers by harnessing their historical records. A budget constrained worker selection model is proposed that focus on learning about the workers skill and effort. Table 5 showcases the relationship between the different constraints.

9 PRIVACY PROTECTION

9.1 Introduction

In most cases, the worker is required to disclose her location information to the third-party SC-server for task assignment or response verification regarding the visit to the task location. However, this shared location information is highly sensitive and susceptible to privacy attacks from adversaries like a potentially untrustworthy SC-server, leading to privacy concerns from workers. Similarly, a task location published online or shared by many workers would result in a privacy breach. With the knowledge of workers' location information, the adversaries can infer the sensitive data of workers like their health information based on the visits to specific hospitals, religious preferences based on the visits to temple/mosque, lifestyle preferences based on their visits to different leisure places. Even in some cases, where the worker uses a fake identity, the location information could still be used to identify the movement patterns of the worker revealing details like the worker's home and workplaces [52], which in turn can be used to reveal her identity.

The privacy concerns of the workers should be addressed by ensuring the location privacy is not violated by the SC system. Otherwise, the workers may refuse to participate in the SC process resulting in failure of SC. Therefore, workers' and tasks' location information should be protected while publishing tasks in different modes.

In the case of WST task publishing mode, the worker does not share her identity and location. However, she is still vulnerable to privacy risk during the task completion and reporting phases. For instance, the worker needs to travel to the task's location during a particular time period to perform the task. With this knowledge, a potential adversary can stalk the workers by creating fake tasks. For example, in [118], the adversaries generated fake accidents to stalk users. Pseudonymity techniques [96] or exchange-based techniques [130] are used as countermeasures to disassociate one's identity with the uploaded data or to increase uncertainty by merging location information of different workers.

Most of the works related to privacy preservation focus on the SAT publishing mode, as all the workers share their location information with the SC-server for task assignment. Cloaking [60, 87], differential privacy-based [104, 105, 131], and encryption-based [95, 96, 137] techniques are used to to protect location privacy of the workers during task assignment. As the name suggests, encryption-based approaches hide the identity and location of workers through encryption. For instance, [95] proposed a secure task assignment strategy to protect privacy in the SAT task publishing mode.

9.2 Cloaking Techniques

With the use of cloaking techniques, the workers' locations are hidden in cloaked regions. The most used form of cloaking in the SC literature is spatial k-anonymity that generates a cloaked region for each worker containing $k-1$ other workers. In [60], a privacy framework named PiRi (Partial-inclusivity and Range independence) is proposed that preserves privacy during task assignment. The framework assumes that the workers trust each other and do not reveal sensitive information about each other. It considers the SC-server as an adversary and workers cannot share their locations. The worker cloaks her location with $k-1$ nearest peers and sends the range queries that are again cloaked among $k-1$ peers to avoid range dependency leaks. The range queries are formed utilizing the maximum radius among all the k peers inside the cloaked region. The framework tries to minimize the number of queries submitted to the SC-server to avoid all-inclusivity leaks. No constraints are considered in this framework.

G-STAC (Global optimization - Spatial Task Assignment with Cloaked locations) and L-STAC (Local optimization - Spatial Task Assignment with Cloaked locations) methods proposed in [87] consider the maximum travel distance of the worker (spatial constraint) while preserving privacy using spatial k-anonymity techniques. Privacy guarantee of k-anonymity based techniques depends upon the specified k value. Choosing an appropriate k value could be difficult as the frequency of workers' visits is not considered. Consequently, the likelihood of an attack on the worker with the most number of visits to a location would be higher. Moreover, k-anonymity based techniques are prone to homogeneity attacks [79] when all the k workers are present at the same location.

9.3 Differential Privacy-based Techniques

Differential Privacy-based techniques refer to distortion of the original location information of the workers by addition of artificial noise. Differential Privacy (DP) [33] addresses participant concerns regarding the leakage of personal information. DP ensures that the released results would not be affected even if the participant removes her data from the input data set. Therefore, an adversary cannot guess whether a participant has participated in the database or not. It would be difficult for an adversary, even with prior knowledge, to infer the data about an individual from the DP's published sanitized data. DP is the most used strategy to protect workers' locations during task assignment [104–106, 131], that addresses the above-mentioned issues of cloaking techniques like prone to homogeneity attacks and privacy guarantee dependency on k value.

[105] proposes a DP-based privacy framework for SC that performs privacy-aware spatial task assignment. The workers do not trust the SC-server to share their location and identity information. However, they will share their locations to the trusted Cellular Service Provider (CSP). CSP collects the locations and releases a private spatial decomposition (PSD) according to the privacy budget ϵ, that was agreed upon by the workers. The CSP can disclose the location information according to the terms agreed with the subscribed workers. The SC-server receives tasks and accesses the PSD to construct a geocast region (GR) that contains sufficient workers such that the queried task is accepted with high probability. The SC-server initiates a geocast communication [84] process to disseminate task t to the workers in the GR. In this case, the SC-server is not allowed to directly establish a contact with the workers inside the GR as it would result in identifying whether a contacted worker is real or fake. Therefore, the communication could be either done through CSP for all the workers or the CSP contacts one of the workers present in the GR and the message would be conveyed on a hop-by-hop basis to the remaining workers in the GR.

Considering privacy concerns complicates the task assignment scenario and reduces the effectiveness and efficiency of the assignment strategies. As the PSD contains false data, there might be scenarios where the geocast region does not contain any workers, and the task has been again queried against the PSD resulting in significant overhead.

The techniques proposed in [105, 131] are based on the scenario where the worker's location does not change, however, in often cases the worker location changes based on her dynamic movement. To enquire the latest location of the workers, SC-server has to request a new PSD release to perform task assignment. However, disclosing multiple versions of the sanitised PSD would be vulnerable to attacks. Conversely, if the PSD increases the noise on every subsequent release, then the data could become potentially useless as SC-server would continuously request for the latest PSD release. [104] tackled this problem by investigating privacy budget allocation techniques across consecutive PSD releases.

The previous studies utilizing the DP-based privacy-preserving task assignment frameworks assume a trusted entity to sanitize the location data [105, 131]. As there is no explicit trust relationship between any two parties in SC, a broader privacy setting where all SC parties are curious,i.e. they learn as much as possible about the inputs, but not malicious would be more appropriate. [108] proposes a three-stage privacy-aware framework, SCGuard, that protects workers and tasks locations without assuming any trusted entity. In the first stage, the SC-server selects a set of potential candidate workers for a given task based on the proximity (calculated based on the perturbed locations), and forwards them to the requester. In the second stage, the requester identifies the most reachable worker from the given set of workers sent by the SC-server and sends the task location to the identified worker. In the third and final stage, the selected worker would accept or reject the received task based on the task's location, i.e., whether the task lies inside the preferred region of the worker. The SCGuard framework assumes a semi-honest adversary model. However, in real-world scenarios this model might not hold as the requesters intent can be malicious, for example, requesters can fake tasks to estimate the workers' locations. Similarly, this approach focuses on the task assignment at a single time instance instead of considering the dynamic nature of workers and tasks. Furthermore, this work assumes that the spatial task does not require multiple workers for completing the task. However, there are many types of tasks that require responses from multiple workers to ensure quality, for instance, reporting the traffic at a road junction.

9.4 Encryption Techniques

Encryption techniques refer to encoding the location information of the workers and tasks so that it can be decoded only by authorized entities with a decryption key. [96] proposes an encryption-based approach to protect workers, identity and location. The *Task service* utilizes onion encryption to hide the worker's identity and location through the *anonymizing network*, Tor[9]. Though the approach is designed to ensure workers' anonymity, there is a possibility for attacks like end-to-end timing correlation attack, as Tor does not try to protect against an attacker with access to the incoming and outgoing traffic of the Tor network. The *Task Service* can perform a timing attack on the worker's location by linking multiple task requests. To prevent such attacks, the workers connect to *Task Service* at random intervals.

In [96], the authors have only focused on the WST mode, where the server has very little control over the task assignment. [95] proposes an encryption-based task assignment approach to protect workers' location privacy in the SAT mode. The proposed privacy framework utilizes a semi-honest third party - *Privacy Service Provider(PSP)* and collects encrypted data from workers along with the encrypted location information. The SC-server performs the worker-task matching by communicating with the PSP in the encrypted domain. The worker close to the task and with a high degree of interest to the task would be chosen. The framework is not relying on a trusted third party as in the case of [105] and is robust to semi-honest adversaries. However, the semi-honest adversary model may not always hold in the real-world scenarios due to its limited privacy protection, as the SC-server and PSP might not follow the specific protocol or the requesters can be malicious.

Similarly, in [75], the locations of workers and tasks are protected by homomorphic encryption. This encryption-based task assignment approach utilizes the worker-task distances computed from the encrypted data. After assignment, the workers receive the encrypted task locations and decrypt them to obtain the exact locations. However, as with all the encryption-based approaches, the overhead computation cost is high when compared to cloaking and DP-based techniques. To improve the overhead computation cost, [74] proposed a strategy to eliminate some complex operations to achieve privacy protection with acceptable overheads. [74] combines partially homomorphic encryption schemes to efficiently perform complex task assignment operations with encryption data.

[9]http://www.torproject.org/

Table 6. Comparison of privacy protected task assignment protocols with multiple constraints

	Spatial	Quality	Budget
Privacy	STAC [87], Hu et al [51]	SUR Protocol [123]	Zhang et al [131], QOI [130]

9.5 Impact on different constraints

While protecting the worker and task location privacy, the SC-server loses the ability to gain knowledge about individual workers for making better task assignments through a learning and optimization process. However, some of the constraints like travel budgets [87] and rewards [130, 131] could be incorporated while assigning tasks to workers (see Table 6). It can be observed that there is very little literature available in SC that combines quality constraints and privacy protection approaches. This is because it is very difficult to assure a certain level of quality from the task outcomes, due to the lack of information about the workers. The approaches to privacy protection and task's quality constraints are negatively impacted by each other as privacy-protected SC might hinder the quality assessment of the individual responses and vice versa. For example, if a task requires a worker with a minimum reputation score and certain skills to be assigned, then the SC-server should be able to know the lower bounds of individual workers' reputation scores and their skillset. However, in a privacy-enabled setting, the SC-server does not possess the knowledge of individual workers needed to assure the requester that the task is assigned satisfying the quality constraints. For example, in [137] the SC-server in the cloud broadcasts the requested task to all the workers, as the workers' information is encrypted, thus reducing the control exerted by the SC-server to assure a certain level of quality.

However, [123] proposes a Secure User Recruitment (SUR) protocol to assure a certain level of quality while protecting the workers' privacy. SUR does not depend on encryption/decryption operations or any other trusted third-party servers. SUR is based on a secret sharing scheme between the different workers. The recruitment decision lies with the workers as the computations are performed at the worker's end. This method incurs a huge computation and communication cost between the workers during the computation process. Furthermore, the SC-server has to broadcast the tasks for all the workers, who would then jointly perform the recruitment.

9.6 Protecting Task and Requesters' Location Privacy

The majority of the current SC literature focuses on protecting the workers' location privacy. However, it ignores the need to protect the task and the requester's location privacy. With task locations being public, there is a privacy risk as the task locations can indirectly reveal the requesters' location. For example, if a requester is posting tasks in the same area, then chances are that the requester's home or workplace would be located in the same area. In [74, 75] the tasks' and requesters' locations are protected through an encryption-based privacy preserving framework. Similarly, [108] protects tasks' and requesters' location privacy though perturbation-based privacy preserving framework.

10 TRUTH DISCOVERY AND CROWDSOURCED DATA AGGREGATION

10.1 Introduction

Truth discovery [43, 72] in SC involves identifying trustworthy information from the responses received from the workers. Truth discovery is relevant when a task is performed by multiple workers, wherein every worker provides uniform answers or conflicting answers. For example, if the task involves identifying the crowd situation in a restaurant at a particular time, there is a high likelihood of receiving differing answers from the workers. The worker might intentionally provide a wrong answer to complete the tasks for earning rewards, or to accumulate points. Sometimes, the differing answers could be unintentional, like the worker has answered about the crowd situation at a different time of the day, which is not relevant to a task. Similarly, some of the spatial tasks are of qualitative nature and require subjective responses. As subjective responses are dependent on the perception of the workers, all of the workers might be answering correctly according to their background. For instance, the task of labelling the crowd at a supermarket might evoke different responses which are all "correct" in the minds of the workers. The workers from a village background might find the supermarket to be crowded, whereas the workers from a city background might classify it as normal. In such cases, identifying the truthful outcome is of utmost importance. All the responses from the workers are aggregated into a single value before relaying the information to the task requester. There are numerous methods in CC to model the worker's reliability and the quality of the aggregated value of the collected task responses [53, 90]. Some of the methods are non-iterative in nature like Majority Voting [65], Honeypot [68], and Expert Label Injected Crowd Estimation (ELICE) [63]. Other methods are of iterative nature like Expectation Maximization [54], Generative Model of Labels,Abilities, and Difficulties (GLAD) [120], Supervised Learning from Multiple Experts (SLME) [89], and Iterative Learning (ITER) [57].

10.2 Non-Iterative Aggregation

The non-iterative aggregation techniques compute a single aggregated value for each question utilizing heuristics like the most common answer. In this section, we will discuss three common non-iterative aggregation techniques, namely Majority Voting, Honeypot and ELICE.

Majority Voting: Majority Voting aggregates to the most recurring specific value among the crowdsourced responses [65]. This aggregation technique does not have any preprocessing involved and does not consider the workers' expertise. Therefore, in the case of potential spammers, the technique might produce an incorrect outcome.

Honeypot: Honeypot aggregation technique is similar to *Majority Voting*, with an additional preprocessing step for detecting fraudulent workers [68]. A set of questions with known answers is randomly merged into the original question set to set up a trap for cheaters. The workers who fail to answer a predetermined number of the added questions (with known answers) will be considered as frauds and removed from the worker set. The probability of each possible value will then be computed with the remaining workers based on the *Majority Voting* technique. However, the success of this technique depends on the set of questions with known answers. In cases, where such questions are not available or if the answers to questions are of a subjective nature, then there is a chance for incorrectly identifying the frauds.

Expert Label Injected Crowd Estimation(ELICE): ELICE is similar to *Honeypot*, except that the question set with known answers would also be used to estimate the expertise level of each worker and the difficulty level of each question [63]. As each answer is weighted by worker expertise and question difficulty, it performs better than the *Majority Voting* and *Honeypot*. However, it has the similar disadvantages as *Honeypot*, with success dependent on the question set with known answers.

10.3 Iterative Aggregation

The iterative aggregation techniques consist of a sequence of iterations, that computes probabilities of answers for each question in each iteration until convergence [54]. In this technique, the set of questions with known answers are not required. In this section, we will consider the four iterative aggregation techniques: EM, GLAD, SLME, and ITER.

Expectation Maximization(EM): The EM technique simultaneously estimates all the probabilities of answers for each question iteratively in two steps: expectation and maximization [54]. In the expectation step, the probabilities of answers for each question are estimated according to the current estimates of their expertise. In the maximization step, this technique re-estimates the expertise of workers based on the current probability of the answer for each question. The process will stop after the probabilities of answers for each question are unchanged in subsequent iterations. Due to the possibility of numerous iterations for reaching convergence, cost of execution might be an issue. Furthermore, EM provides a locally optimal solution rather than a globally optimal one. [28] improves the EM method using pruning and search-based approach to provide a globally optimal solution.

Generative Model of Labels, Abilities, and Difficulties (GLAD): This technique is similar to EM, except that it estimates toughness of the question along with workers' expertise [120] like the non-iterative aggregation technique ELICE [63]. The technique tries to address two special cases: a worker with higher expertise is more likely to answer a question correctly, and a question with higher difficulty has a lower probability of being answered correctly. The initial values of the worker expertise and question toughness impacts the performance of this technique.

Supervised Learning from Multiple Experts (SLME): This technique is similar to EM, however, instead of a confusion matrix, it characterizes the worker expertise by *sensitivity*, which is the ratio of positive answers that are correctly answered and *specificity*, which is the ratio of negative answers that are correctly answered. Due to the limitation of binary labeling for sensitivity and specificity, the SLME technique is incompatible with multiple labels.

Iterative Learning (ITER): This technique [57] is based on standard belief propagation. Similar to GLAD [120] and ELICE [63], ITER also estimates the toughness of the question and the worker expertise. However, unlike the other aggregation methods which assume the reliability of all answers provided by the worker as a single value, ITER computes the reliability for each answer separately. Similarly, for each worker, the toughness of a question is computed. The worker expertise is estimated as the sum of the reliability of her answers weighted by the toughness of answered questions. Unlike the other aggregation techniques, the initial values of answer reliability and question toughness do not impact the performance of ITER.

10.4 Truth Discovery in Spatial Crowdsourcing

However, the current literature in CC do not consider the spatial attributes of the workers and the tasks. They are not effective when both the worker's reliability and her mobility is uncertain. Moreover, there are other spatial attributes that should be considered for truth discovery, like distance to the task from the worker's location [50], task location popularity/influence, location visit tendencies [85]. [50] proposed a probabilistic graphical model for truth inference in crowdsourced POI labelling. The inference model utilizes the location data of the workers and the POIs, along with the POI location influence, and the worker's reliability. The estimated worker reliability and the POI influence would be utilized by the task assignment module to recommend relevant POIs to workers while ensuring quality. Based on the responses collected from the assigned tasks, the inference model will update the worker's reliability information. However, this model does not require the workers to be physically present at the POIs in order to label them. Furthermore, it does not consider the worker's individual visit information to the POI locations to ascertain the trustworthiness of the worker.

The model proposed in [50] requires continuous tracking of the location of the worker, which could lead to a privacy issue. To address this, [85] proposed a probabilistic graphical model, Truth Finder for Spatial Events (TSE), that does not require continuous location tracking of the workers, it instead considers location popularity and the individual worker historic visit indicators to the locations of tasks, in addition to the worker's reliability. However, estimating location popularity does not resolve the issue whether an individual worker would visit that location or not. Hence, an improved model, Personalised Truth Finder for Spatial Events (PTSE), based on personal location visit tendencies of workers is proposed. Similar to [85], [45] has proposed a truth inference algorithm based on the Bayesian estimation model. This model considers the state of the event or task at different time instants based on the received worker responses.

11 APPLICATIONS

As discussed earlier, SC paradigm has the potential of solving many real-world problems like collecting motion traces from building inhabitants to construct floor-plans [3]. According to a survey conducted on crowdsourcing systems in 2011 by Yuen et al, [128], we can classify the crowdsourcing applications into the following four groups: Voting System, Information Sharing Systems, Game and Creative System. Inspired from this classification categories, we have grouped the SC applications into three broad categories based on the utilization of sensors, human knowledge, and human efforts: data collection, query answering, and personal service. Generally, the tasks involving quantifiable information falls under the data collection category and the ones involving qualitative information falls under the query answering category.

11.1 Data Collection

Data Collection refers to the process of gathering information from the different sensors of workers' smartphones at particular locations, without utilizing the workers' knowledge capabilities. Most of the applications belonging to this category are labelled with "mobile crowdsensing" instead of SC. To describe the data collection process, let us take an example of building indoor floor plans from the traces of movement by the workers with the help of their smartphones [3]. These motion traces based on the inertial sensors in the workers' smartphones, which are collected and processed later to build accurate indoor floor plans. Similarly, there are applications to collect data about the public transport drivers' routing behaviour to detect traffic anomalies [86]. A different version of the public transportation crowd-sourced routing algorithm is proposed in [24]. Furthermore, there are applications to collect the accelerometer data to detect the movement of vehicles in a parking lot to determine the parking availability [83].

Many SC applications concentrate on post-processing of the collected data, to improve the quality of the collected data or to generate meaningful data from the raw data. For example, in [3], the collected raw sensor data is processed to generate accurate motion traces of the users. Similarly, post-processing is employed to improve the geo-spatial linked open data ([56], [55]). Post-processing could be performed in real-time [8, 86] or after the completion of data collection phase [3]. The decision to choose a real-time post-processing option would be based on the objective of the SC application.

11.2 Query Answering

Query Answering is another class of SC application that involves harnessing the worker's knowledge to answer a group of questions, related to a specific location or region. Unlike the case of data collection, the information gathering process is not just limited to collecting sensor data of workers' smartphones. Query Answering also utilizes the worker's skill and the ability for answering the queries/tasks. For example, CrowdHelp [8] utilizes query answering concept to improve emergency response for patients during a disaster. The system helps workers assess a patient's physical condition and symptoms, through a series of queries. Similarly, high-quality maps are generated based on the inputs from the crowd [13].

The query answering applications can be further divided based on the type of queries answered. Some of the different types of queries are simple binary (Yes/No) queries, multiple options queries, tagging the images with relevant tags, categorizing different images, describing images, and classifying the type of a spatial feature [117].

Applications consider spatial constraints while collecting the answers from the workers. For example, the option of answering a query is only visible to the workers in the vicinity of the question's geographical location [46]. Similarly, the answers collected are prioritized according to the proximity of the worker to the queries' location. For instance, in [18], the priority was given to the answers provided by the identified local experts among the Twitter users.

11.3 Personal Service

Personal Service is another class of SC application that involves an additional human effort to perform the task, like pick-up and delivery of a package/groceries/food order[10], taxi calling and ride sharing[11], performing a task like painting/cleaning/lawn mowing [12]. The personal service applications have to consider the different spatio-temporal, budget and quality constraints for solving the task matching and task scheduling problems. For example, consider the Ubereats[10] application, involves collecting the food packages from the restaurant and delivery to the customers. The Ubereats SC application matches the food delivery tasks with the worker and plans the route for delivering them.

11.4 Example SC applications

In this section, we choose a representative set of applications, including Uber[11], TaskRabbit[12], and gMission[14], and discuss their features and relevance to SC.

Uber: is an (unlicensed) taxi calling and a peer-to-peer ride-sharing SC application, wherein passengers attempt to hire a taxi or share a ride. It belongs to the personal service class of SC applications. During the operation, Uber's back-end server matches the service requesters and service providers, i.e., drivers, in accordance with their spatiotemporal proximity. For taxi-calling service, the server matches the passenger to the nearest available drivers, according to their selected taxi type. For carpooling services like *UberPool* ride-sharing service, the server matches the potential passengers, who are willing to share the ride to the nearest available driver according to the respective passengers' and driver's source and destination locations. The general optimization target is to improve the service delivery rate and the earnings of the service providers. The server considers constraints like the maximum number of trip requests per day for the driver, pick up time and vehicle choice of the passenger. In particular, constraints such as driver's destinations are further added to ensure that the pickup locations of trip requests will not be far from the destination of the driver.

TaskRabbit[12]: is an online and mobile marketplace, wherein requesters can post tasks that can be performed by verified workers. It belongs to both the query answering and personal service class of SC applications. TaskRabbit services tasks like cleaning the apartment, picking up and delivering the groceries from the supermarket, researching for a party, handyman work. Based on the description of the task provided by the requester, TaskRabbit matches the tasks' required skills with the workers' specified skills within a neighborhood and provides a list of qualified workers for the requester to choose according to their hourly rates. For suggesting workers to the requester, TaskRabbit utilizes both the worker skills, that are verified through a vetting process and worker reputation, based on the feedback from the past tasks.

gMission[14]: is an open-sourced, general purpose SC application that supports a variety of spatial tasks. The application belongs to both the data collection and query answering classes of SC applications. gMission offers credits as an incentive for performing a task. For a newly registered

[10] www.ubereats.com
[11] www.uber.com
[12] www.taskrabbit.com

user to request for a task, one has to spend the credits earned through performing tasks. gMission uses *K-nearest neighbors* algorithm for task assignment, utilizes *Majority Voting* for response aggregation, and improves task workload distribution among workers through dynamic weights ascertained by previous assignments. However, gMission do not consider worker expertise/skill for assigning tasks to workers.

Recently, there is a massive surge in SC applications servicing different purposes. However, the majority of the current SC applications do not conform to the latest developments in SC literature for the deployment of more advanced applications. For instance, except for a few applications like Uber, the majority of existing SC applications do not recommend workers to move to a different place for more tasks, based on the historical information [129]. Also, general SC applications like gMission can improve the user participation by including an incentive mechanism such as monetary rewards for tasks with respect to the levels of workers' expertise or reputation [27, 107]. Moreover, privacy concerns can be addressed for existing SC applications [37], e.g., gMission, so that the sensitive identity or location information is not leaked during the running of applications. Furthermore, scheduling mechanisms [73] can be incorporated in SC applications, such as gMission and TaskRabbit, for enhancing the task completion ratio, if the user participation ratio is high.

12 DISCUSSION

Although research is gaining momentum in SC, it is still in the nascent stage with a lot of open research issues. In this section, we will discuss the different challenges and limitations of existing research work. These challenges are related to task matching and scheduling problems, truth inference models, privacy issues, and the lack of real-world datasets.

12.1 Task Matching and Scheduling issues

The task matching and scheduling issues are related to the dynamic arrival of tasks and workers, the optimization goals, the immutability constraint, and the workers' movement patterns.

The majority of the work done in SC does not account for the dynamic arrival of workers and tasks to the SC-server during the task assignment phase [15, 26, 61] or task scheduling phase [29]. They work on the assumption that the SC-server possesses the complete knowledge of the inputs sets of tasks and workers. This renders their solutions inefficient in dynamic real-time environments. Recently, some solutions were proposed to address the task matching problems in online scenarios like [47, 103, 113], however, only tasks are considered to be dynamic. In [111], the authors proposed a solution to factor the dynamic nature of both workers and tasks arrival. However, [111] does not provide support for constraints like quality and privacy.

The current task matching approaches do not consider the "workplaces" [97] aspect of SC. For instance, InterestingSport[13] is a SC application that helps users to find suitable trainers and book sport facilities in real-time. Therefore, in addition to checking the availabilities of the users and trainers, the SC-server has to take into account the availability of the sport facilities. [97] formulates a Trichromatic online matching in real time SC (TOM) problem that considers workplaces along with the tasks and the workers during the assignment. The "workplaces" extension could be easily accommodated in the existing worker-task matching framework by utilizing the "workplaces" information as additional constraints to the worker-task matching problem. However, further analysis should be done to ascertain the impact this additional constraint has on existing constraints like privacy, budget and quality.

[13]"InterestingSport," http://www.quyundong.com/

Similarly, [73] proposes solutions to schedule dynamically arriving tasks to a single static worker. There is a lack of research for scheduling dynamically arriving tasks for multiple workers. [30] tries to address this issue by considering a predicted set of tasks and worker. An iterative strategy is proposed for sequentially assigning a set of tasks to workers and scheduling the assigned tasks to each worker. However, this strategy does not consider the compatibility of a task to the worker's schedule before assignment, thereby resulting in an overhead of re-assigning the unscheduled task to a different worker.

The optimization goals for the different task matching problems in the SC-literature are focused on benefitting either the worker or the task. If the SC-server assigns tasks to workers based on the preferences of tasks, then it would benefit the tasks. For example, assigning tasks to workers that have a minimum reputation [62] or a given set of skills [107]. Similarly, if the SC-server assigns tasks to workers based on their preferences, it would benefit the workers. For example, if the SC-server tries to maximize the reward received by workers [27]. Benefitting either the tasks or the workers would result in inefficient assignment for the other. For example, the workers may fail to find tasks that they would like to perform and the SC-server may fail to find workers that provide better quality responses. Therefore, to address this dilemma, it would be beneficial if both the workers and tasks preferences are taken into account in the task assignment scheme.

Almost all the solutions proposed to solve the task matching problems follows the rule of immutability for task assignments, i.e., when a task is assigned to a worker, it cannot be revoked. The immutability constraint is introduced to prune the search space by removing the assigned tasks. Although this constraint reduces the search space, there are cases where this rule might not be beneficial. Some of these cases are the online scenarios with the objectives of improving task acceptance rate and quality of task responses. For example, a new task with better utility/reward or a new worker with better reputation/skill might be available after assignment. Therefore, revoking the immutability constraint on task assignments would help in re-assignment of tasks to improve the reward/utility received, leading to a higher task acceptance. Furthermore, the overall quality of task responses by the workers would be improved.

Furthermore, workers tend to accept tasks close to their commute. Therefore, considering the workers' movement patterns while assigning tasks [12] would improve the task acceptance rate. However, there are a lot of open issues regarding the utilization of the workers' movement patterns for task assignment. For instance, the worker route is not dynamically updated, even if there is a better task available with higher utility, i.e., less travel cost and higher pay. Similarly, with the existing strategies, there is a possibility that some workers monopolize the task assignments with their willingness to travel long distances, resulting in the reduction of worker diversity.

12.2 Privacy Issues

Privacy concerns are one of the most fundamental problems of the worker. Although some literature in SC addressed this issue (as discussed in Section 9), there are a lot of open issues that need to be addressed. Privacy concerns are fuelled by the workers' lack of trust on the third party SC-server. To address this, some solutions utilize the concept of differential privacy to get the workers aggregated data from worker-trusted cellular service providers [105], to anonymize workers from the SC-server. However, the SC-server would still have the knowledge of the task locations and the time interval during which the assigned worker would visit the task location, resulting in a serious privacy breach. Furthermore, by anonymizing the worker to the SC-server, the support for individual spatial constraints and quality constraints is hindered with the existing strategies. Although [123] deals with the privacy-enabled quality assurance problem, it do not consider the differing requirements of the tasks and the differing spatial constraints of the workers. Furthermore, the worker's reliability scores cannot be updated based on the success/failure of the assigned tasks in [123].

12.3 Truth Inference Models

In the existing truth inference models of SC, it is difficult to infer truth while considering worker's location privacy. For example, this is the case in location obfuscation, wherein the location of the worker is generalized to protect the exact location of the worker. In such cases, the existing truth inference models [85] simply ignore the workers with obfuscated locations. However, considering the importance of individual workers' location privacy, new models are needed to deal with these issues.

Similarly, the current truth inference models of SC assumes that the workers respond independently of each other, i.e., there would not be any copying or sharing of answers among the workers. However, in reality, there could be a case of copying among the workers, that could result in improper estimation of worker reliability, thereby affecting the quality of the task responses. Therefore, copy-detection methods are needed while inferring truth from the worker responses [31]. To prevent collusion between workers, it is important to quantify the probability that the workers collude with each other based on the quality of their responses to better assign the tasks. For example, in CC literature, [101] has proposed a three-step based *θ-secure task assignment approach* for task assignment avoiding collusion between workers. Similar approaches are needed in SC considering the spatial characteristics of workers and tasks. Furthermore, new approaches should be proposed similar to [28] for providing a global optimal solution in estimating the task ground truth and worker expertise in SC.

12.4 Lack of real world datasets

The primary challenge faced by all the solutions proposed in the SC literature is to evaluate the strategies based on real-world datasets. Due to the lack of publicly available real-world datasets, SC algorithms are evaluated utilizing synthetic datasets, that are generated based on different distribution functions. Some SC works utilize few real-world datasets related to location-based social networks like Gowalla[14] and Bright-kite[15], where users can check-in to different points of interest in their vicinity. These location-based social networks datasets are adapted to the SC scenario by assuming the check-in spots as task locations, users as workers and a user checking into a spot is considered as accepting the task [61, 134]. To et al. [102], advanced this strategy to generate synthetic SC datasets with realistic spatiotemporal properties and constraints adapted from geosocial datasets like Gowalla and Yelp [16]. The advantage with these datasets is that they exhibit the workers nature of preferring to perform nearby tasks. However, there might be different geosocial phenomena in SC that are not observed with either synthetic or adapted datasets. There is a need to design a SC platform to collect real-world data for researchers to advance the research in SC.

12.5 Lacking User Participation

For an SC application to be successful, it should be able to attract many task requesters and workers. Most of the existing applications are based on voluntary participation, and as performing tasks involve spending time, effort and resources such as smartphone's memory and battery, it is difficult to attract workers without offering rewards. Moreover, with the privacy concerns, the users might not be willing to participate in the SC application. To motivate users to participate in SC, incentive mechanisms are developed that involve monetary rewards [27, 48, 67] and virtual credits [14, 66]. Though incentive mechanisms have a positive impact on improving the user participation, it is still

[14]http://snap.stanford.edu/data/loc-gowalla.html
[15]http://snap.stanford.edu/data/loc-brightkite.html
[16]http://yelp.com/dataset_challenge

limited to the users who are aware of the SC paradigm. To harness the true potential of SC, new methods should be developed to enable the general public to become aware of the SC application and to convince them to participate in SC applications.

13 FUTURE RESEARCH DIRECTIONS

In this section, we provide some of the potential research directions in SC, based on the discussion in Section 12.

Improving Task Assignment Protocols in Online Scenarios:

As discussed in Section 12, task assignment protocols should be improved to tackle the uncertainty of the dynamic real-time SC environment. Although, different solutions like [111] are proposed to address this, further research needs be performed for improving the efficiency and for ascertaining the impact on different constraints like quality and privacy. In particular, strategies should be proposed to perform task assignment in an online scenario with privacy-enabled SC.

Furthermore, as discussed in Section 12, the existing task assignment strategies are restricted by the immutability constraint on task assignments. However, in uncertain online scenarios, it would be beneficial to revoke this constraint and check for all the suitable tasks and workers. There would be drawbacks for revoking the immutability constrain, like the revoked task might not find a suitable worker afterwards and could remain unassigned or expire before being assigned, the worker might have started the travel to the current task location, wherein changing the assignment would cause confusion and dissatisfaction. However, if the workers and SC-server has a prior agreement regarding the potential change in assignments and a set of pre-requisites are agreed upon, like a threshold time for updating tasks, then the immutability constraint could be beneficial and practical. Correspondingly, strategies should be proposed allowing the update of assigned tasks based on the available tasks and workers respecting the different constraints like deadlines and required expertise.

Assignment Protocols that Benefit Both Workers and Tasks:

As discussed in Section 12, the existing task matching protocols either attempt to assign tasks to workers based on workers' preferences or tasks preferences. Optimizing the benefits for both the tasks and the workers would improve the success of the assignment protocol. A new assignment strategy could be proposed by adapting the solution proposed in the CC literature [136] with spatiotemporal context. Zheng et al, [136] has proposed a task assignment framework in CC, called Task Assignment with Mutual Benefit Awareness (TAMBA), to offer mutual benefit to workers and tasks, based on their preferences extracted from the historical data. Similarly, [5] have proposed an auction-based framework, Auction-SC, that would benefit both workers and the SC-server by allowing workers to bid on the arriving tasks.

Integration of task publishing modes:

There is a need for an effective framework that combines both task publishing modes (WST and SAT), to provide effective solutions for improving the efficiency of SC and the task acceptance rate. For instance, the workers can select their preferred tasks via WST mode. The case where multiple workers are opting for the same task, the SC-server can employ the SAT mode to resolve the conflict.

Privacy-enabled Truth Inference Models:

The current truth inference models in SC, can avoid real-time location tracking of the workers and exploit the historical information for inferring truth and to worker reliability models. However, in a privacy-protected SC, the individual worker locations would be unknown to the SC-server, which the current inference models cannot support. Therefore, new truth inference models should be proposed for the different location privacy models to ensure quality to the task requester. Furthermore, new techniques are needed to process complex textual or multimedia information to assess the trustworthiness of the responses [90].

Quality Assurance-Privacy Preservation Trade off:

As discussed in Section 9.1, privacy preservation and quality assurance negatively affect each other as privacy-protected SC might hinder the quality of the responses and vice versa, especially in the case of single response tasks with differing requirements. Therefore, a trade-off mechanism is needed to balance the privacy requirements of workers and quality constraints of tasks set by requesters. For instance, with some cloaking techniques like k-anonymity and quality constraint information like reliability and expertise skill set can be aggregated on the client side of the workers. An aggregate query of k workers can be sent to SC-server along with the cloaked region and quality constraints information for task assignment. Furthermore, the location information of tasks should also be protected from the SC-server along with the worker information to avoid privacy breach.

Harnessing Worker Movement Patterns:

Mining workers' movement patterns would help us in predicting workers' movement behavior and her availability for performing tasks. The predicted information about workers' movement would allow the SC-server to know about the workers' arrival order, thereby reducing the uncertainty in the online task assignment scenario. The outcome of such assignments would be the trajectory of the worker containing the assigned tasks. Furthermore, an additional constraint like the order of the tasks, should be considered during task assignment, if there exists a sequential dependency between the tasks, i.e., task t_A should be performed after task t_B. In [119], a worker mobility prediction model was proposed to align the workers mobility with spatial task requirements for task assignment.

Improving User Participation in SC:

As discussed in Section 12, there is a need to devise better strategies to attract users to improve the participation in SC applications. The existing strategies like incentive mechanisms are useful to an extent. However, they are still limited to the userbase familiar with SC. The new strategies should expand the reach of the SC applications to users who are not so familiar with SC as well.

Harnessing Geo-Social Network Information:

Current SC literature contains little work considering the geosocial relationships between the workers that could be helpful in enriching the worker's profile. Location influence [93] of workers can be utilized to provide a partial ranking to workers in team-oriented task planning [42]. Resulting partial ranking, SC-server can select the leaders for the teams. Furthermore, location influence concept can be used in allocating rewards to the workers in a dynamic budget reward model. workers with the highest location influence ranking would be attracted to perform the task by offering higher reward. By attracting the highly ranked location influencers, their followers are attracted to the task location. Thus, increasing the worker diversity of the location and improving the chances for a task to be assigned in the neighborhood.

14 SUMMARY

In this survey, we reviewed the existing literature related to SC from a technical perspective. We distinguished different topics in the research and proposed our taxonomy to organize them. We noticed that the architecture of SC adapts the structure of CC to serve spatiotemporal interests. Furthermore, we observed that the majority of the existing work focuses on task matching along with varying constraints since the SC-server exerts more control on the task matching proceedings. Similarly, we observed a significant density of research focusing on offline scenarios. Our comparison study revealed the shortcomings of the different strategies and identified relationships between the various constraints of SC. The quality constraints are found to be negatively impacted by the privacy protection approaches and positively correlated with the budget constraints. The identified shortcomings and challenges are related to the task assignment in online scenarios, the dynamic movement of workers, the privacy-quality trade-off, and the geosocial relationships. We suggest future work to address these challenges and advance the application spectrum of SC.

REFERENCES

[1] Mohammad Allahbakhsh, Boualem Benatallah, Aleksandar Ignjatovic, Hamid Reza Motahari-Nezhad, Elisa Bertino, and Schahram Dustdar. 2013. Quality control in crowdsourcing systems: Issues and directions. *IEEE Internet Computing* 17 (2013), 76–81.

[2] Florian Alt, Alireza Sahami Shirazi, Albrecht Schmidt, Urs Kramer, and Zahid Nawaz. 2010. Location-based crowdsourcing: extending crowdsourcing to the real world. In *NordiCHI*. 13–22.

[3] Moustafa Alzantot and Moustafa Youssef. 2012. CrowdInside : Automatic Construction of Indoor Floorplans. In *GIS*. 99–108.

[4] Mohammad Asghari and Cyrus Shahabi. 2017. An On-line Truthful and Individually Rational Pricing Mechanism for Ride-sharing. In *GIS*. 7.

[5] Mohammad Asghari and Cyrus Shahabi. 2017. On on-line task assignment in spatial crowdsourcing. In *Big Data*. 395–404.

[6] Judit Bar-Ilan, Guy Kortsarz, and David Peleg. 2001. Generalized submodular cover problems and applications. *Theoretical Computer Science* 250, 1 (2001), 179–200.

[7] K Benouaret, R Valliyur-Ramalingam, and F Charoy. 2013. Answering complex location-based queries with crowdsourcing. In *Collaboratecom*. 438–447.

[8] Liliya I Besaleva and Alfred C Weaver. 2013. CrowdHelp: Application for Improved Emergency Response through Crowdsourced Information. In *UbiComp*. 1437–1446.

[9] Daren C Brabham. 2008. Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence* 14, 1 (2008), 75–90.

[10] Jeffrey A Burke, Deborah Estrin, Mark Hansen, Andrew Parker, Nithya Ramanathan, Sasank Reddy, and Mani B Srivastava. 2006. Participatory sensing. In *ACM SenSys Workshop, World-Sensor-Web*. 1–5.

[11] Georgios Chatzimilioudis, Andreas Konstantinidis, Christos Laoudias, and Demetrios Zeinalipour-Yazti. 2012. Crowdsourcing with smartphones. *IEEE Internet Computing* 16, 5 (2012), 36–44.

[12] Cen Chen, Shih-Fen Cheng, Aldy Gunawan, Archan Misra, Koustuv Dasgupta, and Deepthi Chander. 2014. Traccs: A framework for trajectory-aware coordinated urban crowd-sourcing. In *HCOMP*. 30–40.

[13] Xi Chen, Xiaopei Wu, Xiang-Yang Li, Xiaoyu Ji, Yuan He, and Yunhao Liu. 2016. Privacy-aware high-quality map generation with participatory sensing. *IEEE Transactions on Mobile Computing* 15, 3 (2016), 719–732.

[14] Zhao Chen, Caleb Chen Cao, Lei Chen, and Chen Jason Zhang. 2014. gMission : A General Spatial Crowdsourcing Platform. In *VLDB Endowment*. 1629–1632.

[15] Peng Cheng, Xiang Lian, Lei Chen, Jinsong Han, and Jizhong Zhao. 2016. Task assignment on multi-skill oriented spatial crowdsourcing. *TKDE* 28, 8 (2016), 2201–2215.

[16] Peng Cheng, Xiang Lian, Lei Chen, and Cyrus Shahabi. 2017. Prediction-based task assignment in spatial crowdsourcing. In *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on*. IEEE, 997–1008.

[17] Peng Cheng, Xiang Lian, Zhao Chen, Rui Fu, Lei Chen, Jinsong Han, and Jizhong Zhao. 2015. Reliable diversity-based spatial crowdsourcing by moving workers. *VLDB Endowment* 8, 10 (2015), 1022–1033.

[18] Zhiyuan Cheng, James Caverlee, Himanshu Barthwal, and Vandana Bachani. 2014. Who is the barbecue king of texas?: a geo-spatial approach to finding local experts on twitter. In *SIGIR*. 335–344.

[19] Man Hon Cheung, Fen Hou, and Jianwei Huang. 2017. Make a difference: Diversity-driven social mobile crowdsensing. In *INFOCOM*. 1–9.

[20] Anand Inasu Chittilappilly, Lei Chen, and Sihem Amer-Yahia. 2016. A survey of general-purpose crowdsourcing techniques. *TKDE* 28, 9 (2016), 2246–2266.

[21] Yohan Chon, Nicholas D Lane, Yunjong Kim, Feng Zhao, and Hojung Cha. 2013. Understanding the coverage and scalability of place-centric crowdsensing. In *UbiComp*. 3–12.

[22] Cory Cornelius, Apu Kapadia, David Kotz, Dan Peebles, Minho Shin, and Nikos Triandopoulos. 2008. Anonysense: privacy-aware people-centric sensing. In *MobiSys*. 211–224.

[23] Justin Cranshaw, Eran Toch, Jason Hong, Aniket Kittur, and Norman Sadeh. 2010. Bridging the gap between physical location and online social networks. In *UbiComp*. 119–128.

[24] To Tu Cuong. 2013. CrowdRoute: A Crowd-sourced Routing Algorithm in Public Transit Networks. In *GIS*. 9–14.

[25] George Danezis, Stephen Lewis, and Ross J Anderson. 2005. How much is location privacy worth?. In *WEIS*, Vol. 5.

[26] Hung Dang, Tuan Nguyen, and Hien To. 2013. Maximum Complex Task Assignment: Towards Tasks Correlation in Spatial Crowdsourcing. In *IIWAS*. 77.

[27] K.-H. Dang and K.-T. Cao. 2013. Towards reward-based spatial crowdsourcing. In *ICCAIS*. 363–368.

[28] Akash Das Sarma, Aditya Parameswaran, and Jennifer Widom. 2016. Towards globally optimal crowdsourcing quality management: The uniform worker setting. In *SIGMOD*. ACM, 47–62.
[29] Dingxiong Deng, Cyrus Shahabi, and Ugur Demiryurek. 2013. Maximizing the number of worker's self-selected tasks in spatial crowdsourcing. In *GIS*. 324–333.
[30] Dingxiong Deng, Cyrus Shahabi, and Linhong Zhu. 2015. Task matching and scheduling for multiple workers in spatial crowdsourcing. In *GIS*. 21.
[31] Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava. 2009. Truth discovery and copying detection in a dynamic world. *VLDB Endowment* 2, 1 (2009), 562–573.
[32] Prabal Dutta, Paul M Aoki, Neil Kumar, Alan Mainwaring, Chris Myers, Wesley Willett, and Allison Woodruff. 2009. Common sense: participatory urban sensing using a network of handheld air quality monitors. In *SenSys*. 349–350.
[33] Cynthia Dwork. 2006. Differential privacy. In *Automata, languages and programming*. Springer, 1–12.
[34] Ju Fan, Guoliang Li, Beng Chin Ooi, Kian-lee Tan, and Jianhua Feng. 2015. icrowd: An adaptive crowdsourcing framework. In *SIGMOD*. ACM, 1015–1030.
[35] Yue Fan, Hailong Sun, Yanmin Zhu, Xudong Liu, and Ji Yuan. 2015. A Truthful Online Auction for Tempo-spatial Crowdsourcing Tasks. In *SOSE*. 332–338.
[36] Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and S Muthukrishnan. 2009. Online stochastic matching: Beating 1-1/e. In *FOCS*. 117–126.
[37] Wei Feng, Zheng Yan, Hengrun Zhang, Kai Zeng, Yu Xiao, and Y Thomas Hou. 2017. A Survey on Security, Privacy and Trust in Mobile Crowdsourcing. *IEEE Internet of Things Journal* (2017), 2971–2992.
[38] Zhenni Feng, Yanmin Zhu, Qian Zhang, Lionel M Ni, and Athanasios V Vasilakos. 2014. TRAC: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing. In *INFOCOM*. 1231–1239.
[39] Zhenni Feng, Yanmin Zhu, Qian Zhang, Hongzi Zhu, Jiadi Yu, Jian Cao, and Lionel M Ni. 2014. Towards truthful mechanisms for mobile crowdsourcing with dynamic smartphones. In *ICDCS*. 11–20.
[40] Raghu K Ganti, Nam Pham, Hossein Ahmadi, Saurabh Nangia, and Tarek F Abdelzaher. 2010. GreenGPS: a participatory sensing fuel-efficient maps application. In *MobiSys*. 151–164.
[41] Raghu K Ganti, Fan Ye, and Hui Lei. [n. d.]. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine* 49, 11 ([n. d.]), 32–39.
[42] Dawei Gao, Yongxin Tong, Yudian Ji, and Ke Xu. 2017. Team-Oriented Task Planning in Spatial Crowdsourcing. In *APWeb-WAIM*. 41–56.
[43] Jing Gao, Qi Li, Bo Zhao, Wei Fan, and Jiawei Han. 2015. Truth discovery and crowdsourcing aggregation: A unified perspective. *VLDB Endowment* 8, 12 (2015), 2048–2049.
[44] Ruipeng Gao, Mingmin Zhao, Tao Ye, Fan Ye, Yizhou Wang, Kaigui Bian, Tao Wang, and Xiaoming Li. 2014. Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing. In *MobiCom*. 249–260.
[45] Daniel A Garcia-Ulloa, Li Xiong, and Vaidy Sunderam. 2017. Truth discovery for spatio-temporal events from crowdsourced data. *VLDB Endowment* 10, 11 (2017), 1562–1573.
[46] Jorge Gonçalves, Denzil Ferreira, Simo Hosio, Yong Liu, Jakob Rogstadius, Hannu Kukka, and Vassilis Kostakos. 2013. Crowdsourcing on the spot: altruistic use of public displays, feasibility, performance, and behaviours. In *UbiComp*. 753–762.
[47] Umair Ul Hassan and Edward Curry. 2014. A Multi-armed Bandit Approach to Online Spatial Task Assignment. In *UIC*. 212–219.
[48] Shibo He, Dong-Hoon Shin, Junshan Zhang, and Jiming Chen. 2014. Toward optimal allocation of location dependent tasks in crowdsensing. In *INFOCOM*. 745–753.
[49] Mahmood Hosseini, Alimohammad Shahri, Keith Phalp, Jacqui Taylor, and Raian Ali. 2015. Crowdsourcing: A taxonomy and systematic mapping study. *Computer Science Review* 17 (2015), 43–69.
[50] Huiqi Hu, Yudian Zheng, Zhifeng Bao, Guoliang Li, Jianhua Feng, and Reynold Cheng. 2016. Crowdsourced poi labelling: Location-aware result inference and task assignment. In *ICDE 2016*. 61–72.
[51] Jie Hu, Liusheng Huang, Lu Li, Mingyu Qi, and Wei Yang. 2015. Protecting location privacy in spatial crowdsourcing. In *APWeb*. 113–124.
[52] Chao Huang, Dong Wang, and Shenglong Zhu. 2017. Where are you from: Home location profiling of crowd sensors from noisy and sparse crowdsourcing data. In *INFOCOM*. 1–9.
[53] Nguyen Quoc Viet Hung, Nguyen Thanh Tam, Lam Ngoc Tran, and Karl Aberer. 2013. An evaluation of aggregation techniques in crowdsourcing. In *WISE*. 1–15.
[54] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. 2010. Quality management on amazon mechanical turk. In *ACM SIGKDD workshop on human computation*. 64–67.
[55] Roula Karam and Michele Melchiori. 2013. A crowdsourcing-based framework for improving geo-spatial open data. In *SMC*. 468–473.

[56] Roula Karam and Michele Melchiori. 2013. Improving geo-spatial linked data with the wisdom of the crowds. In *EDBT Workshop*. 68–74.

[57] David R Karger, Sewoong Oh, and Devavrat Shah. 2011. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems*. 1953–1961.

[58] Richard M Karp. 1992. On-line algorithms versus off-line algorithms: How much is it worth to know the future?. In *Proc. of the Information Processing IFIP Congress (1)*, Vol. 12. 416–429.

[59] Ryoma Kawajiri, Masamichi Shimosaka, and Hisashi Kahima. 2014. Steered crowdsensing: Incentive Design towards Quality-Oriented Place-Centric Crowdsensing. In *UbiComp*. 691–701.

[60] Leyla Kazemi and Cyrus Shahabi. 2011. A privacy-aware framework for participatory sensing. *ACM SIGKDD Explorations Newsletter* 13, 1 (2011), 43–51.

[61] Leyla Kazemi and Cyrus Shahabi. 2012. Geocrowd: enabling query answering with spatial crowdsourcing. In *GIS*. 189–198.

[62] Leyla Kazemi, Cyrus Shahabi, and Lei Chen. 2013. Geotrucrowd: trustworthy query answering with spatial crowdsourcing. In *GIS*. 314–323.

[63] Faiza Khan Khattak and Ansaf Salleb-Aouissi. 2011. Quality control of crowd labeling through expert evaluation. In *NIPS 2nd Workshop on Computational Social Science and the Wisdom of Crowds*, Vol. 2. 5.

[64] Iordanis Koutsopoulos. 2013. Optimal incentive-driven design of participatory sensing systems. In *INFOCOM*. 1402–1410.

[65] Ludmila I Kuncheva, Christopher J Whitaker, Catherine A Shipp, and Robert PW Duin. 2003. Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis & Applications* 6, 1 (2003), 22–31.

[66] Kun-chan Lan, Chien-Ming Chou, and Han-Yi Wang. 2012. An incentive-based framework for vehicle-based mobile sensing. *Procedia Computer Science* 10 (2012), 1152–1157.

[67] Juong-Sik Lee and Baik Hoh. 2010. Dynamic pricing incentive for participatory sensing. *Pervasive and Mobile Computing* 6, 6 (2010), 693–708.

[68] Kyumin Lee, James Caverlee, and Steve Webb. 2010. The social honeypot project: protecting online communities from spammers. In *WWW*. 1139–1140.

[69] Janette Lehmann, Carlos Castillo, Mounia Lalmas, and Ethan Zuckerman. 2013. Finding news curators in twitter. In *WWW(TheWebConf)*. 863–870.

[70] Janette Lehmann, Carlos Castillo, Mounia Lalmas, and Ethan Zuckerman. 2013. Transient News Crowds in Social Media.. In *ICWSM*.

[71] Guoliang Li, Jiannan Wang, Yudian Zheng, and Michael J Franklin. 2016. Crowdsourced data management: A survey. *TKDE* 28, 9 (2016), 2296–2319.

[72] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. A survey on truth discovery. *ACM SIGKDD Explorations Newsletter* 17, 2 (2016), 1–16.

[73] Yu Li, Man Lung Yiu, and Wenjian Xu. 2015. Oriented online route recommendation for spatial crowdsourcing task workers. In *Advances in Spatial and Temporal Databases*. Springer, 137–156.

[74] An Liu, Weiqi Wang, Shuo Shang, Qing Li, and Xiangliang Zhang. 2017. Efficient task assignment in spatial crowdsourcing with worker and task privacy protection. *GeoInformatica* (2017), 1–28.

[75] Bozhong Liu, Ling Chen, Xingquan Zhu, Ying Zhang, and Chengqi Zhang. 2017. Protecting location privacy in spatial crowdsourcing using encrypted data. In *EDBT*. 478–481.

[76] Xuan Liu, Meiyu Lu, Beng Chin Ooi, Yanyan Shen, Sai Wu, and Meihui Zhang. 2012. Cdas: a crowdsourcing data analytics system. *VLDB Endowment* 5, 10 (2012), 1040–1051.

[77] Tom H Luan, Longxiang Gao, Zhi Li, Yang Xiang, Guiyi Wei, and Limin Sun. 2015. Fog computing: Focusing on mobile users at the edge. *arXiv preprint arXiv:1502.01815* (2015).

[78] Tie Luo, Hwee-Pink Tan, and Lirong Xia. 2014. Profit-maximizing incentive for participatory sensing. In *INFOCOM*.

[79] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. 2006. l-diversity: Privacy beyond k-anonymity. In *ICDE*. 24.

[80] Adam Marcus, Aditya Parameswaran, et al. 2015. Crowdsourced data management: Industry and academic perspectives. *Foundations and Trends® in Databases* 6, 1-2 (2015), 1–161.

[81] Winter Mason and Duncan J Watts. 2010. Financial incentives and the performance of crowds. *ACM SigKDD Explorations Newsletter* 11, 2 (2010), 77–85.

[82] J Michael Moore. 1968. An n job, one machine sequencing algorithm for minimizing the number of late jobs. *Management science* 15, 1 (1968), 102–109.

[83] Anandatirtha Nandugudi, Taeyeon Ki, Carl Nuessle, and Geoffrey Challen. 2014. PocketParker: pocketsourcing parking lot availability. In *UbiComp*. 963–973.

[84] Julio C Navas and Tomasz Imielinski. 1997. GeoCast—geographic addressing and routing. In *MobiCom*. 66–76.

[85] Robin Wentao Ouyang, Mani Srivastava, Alice Toniolo, and Timothy J Norman. 2016. Truth discovery in crowdsourced detection of spatial events. *TKDE* 28, 4 (2016), 1047–1060.
[86] Bei Pan, Yu Zheng, David Wilkie, and Cyrus Shahabi. 2013. Crowd Sensing of Traffic Anomalies Based on Human Mobility and Social Media. In *GIS*. 344–353.
[87] Layla Pournajaf, Li Xiong, Vaidy Sunderam, and Slawomir Goryczka. 2014. Spatial task assignment for crowd sensing with cloaked locations. In *IEEE MDM*. 73–82.
[88] Lingjun Pu, Xu Chen, Jingdong Xu, and Xiaoming Fu. 2016. Crowdlet: Optimal worker recruitment for self-organized mobile crowdsourcing. In *INFOCOM*. 5.
[89] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Anna Jerebko, Charles Florin, Gerardo Hermosillo Valadez, Luca Bogoni, and Linda Moy. 2009. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *ICML*. 889–896.
[90] Francesco Restuccia, Nirnay Ghosh, Shameek Bhattacharjee, Sajal K Das, and Tommaso Melodia. 2017. Quality of Information in Mobile Crowdsensing: Survey and Research Challenges. *TOSN* 13, 4 (2017), 34.
[91] Herbert Robbins. 1985. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*. Springer, 169–177.
[92] Jakob Rogstadius, Vassilis Kostakos, Aniket Kittur, Boris Smus, Jim Laredo, and Maja Vukovic. 2011. An Assessment of Intrinsic and Extrinsic Motivation on Task Performance in Crowdsourcing Markets.. In *ICWSM*. 17–21.
[93] Muhammad Aamir Saleem, Rohit Kumar, Toon Calders, Xike Xie, and Torben Bach Pedersen. 2017. Location Influence in Location-based Social Networks. In *WSDM*. 621–630.
[94] Alexander Schrijver. 2002. *Combinatorial optimization: polyhedra and efficiency*. Vol. 24. Springer Science & Business Media.
[95] Yao Shen, Liusheng Huang, Lu Li, Xiaorong Lu, Shaowei Wang, and Wei Yang. 2015. Towards preserving worker location privacy in spatial crowdsourcing. In *GLOBECOM*. 1–6.
[96] Minho Shin, Cory Cornelius, Dan Peebles, Apu Kapadia, David Kotz, and Nikos Triandopoulos. 2011. AnonySense: A system for anonymous opportunistic sensing. *Pervasive and Mobile Computing* 7, 1 (2011), 16–30.
[97] Tianshu Song, Yongxin Tong, Libin Wang, Jieying She, Bin Yao, Lei Chen, and Ke Xu. 2017. Trichromatic online matching in real-time spatial crowdsourcing. In *ICDE*. 1009–1020.
[98] Matthias Stevens and Ellie D'Hondt. 2010. Crowdsourcing of pollution data using smartphones. In *Workshop on Ubiquitous Crowdsourcing*.
[99] Brian L Sullivan, Christopher L Wood, Marshall J Iliff, Rick E Bonney, Daniel Fink, and Steve Kelling. 2009. eBird: A citizen-based bird observation network in the biological sciences. *Biological Conservation* 142, 10 (2009), 2282–2292.
[100] Dezhi Sun, Ke Xu, Hao Cheng, Yuanyuan Zhang, Tianshu Song, Rui Liu, and Yi Xu. 2018. Online delivery route recommendation in spatial crowdsourcing. *World Wide Web* (2018), 1–22.
[101] Haipei Sun, Boxiang Dong, Bo Zhang, Wendy Hui Wang, and Murat Kantarcioglu. 2018. Sensitive Task Assignments in Crowdsourcing Markets with Colluding Workers. In *ICDE*.
[102] Hien To, Mohammad Asghari, Dingxiong Deng, and Cyrus Shahabi. 2016. SCAWG: A toolbox for generating synthetic workload for spatial crowdsourcing. In *PerCom Workshop*. 1–6.
[103] Hien To, Liyue Fan, Luan Tran, and Cyrus Shahabi. 2016. Real-time task assignment in hyperlocal spatial crowdsourcing under budget constraints. In *PerCom*. 1–8.
[104] Hien To, Gabriel Ghinita, Liyue Fan, and Cyrus Shahabi. 2017. Differentially private location protection for worker datasets in spatial crowdsourcing. *IEEE Transactions on Mobile Computing* 16, 4 (2017), 934–949.
[105] Hien To, Gabriel Ghinita, and Cyrus Shahabi. 2014. A framework for protecting worker location privacy in spatial crowdsourcing. *VLDB Endowment* 7, 10 (2014), 919–930.
[106] Hien To, Gabriel Ghinita, and Cyrus Shahabi. 2015. PrivGeoCrowd: A toolbox for studying private spatial crowdsourcing. In *ICDE*. 1404–1407.
[107] Hien To, Cyrus Shahabi, and Leyla Kazemi. 2015. A server-assigned spatial crowdsourcing framework. *ACM Transactions on Spatial Algorithms and Systems* 1, 1 (2015), 2.
[108] Hien To, Cyrus Shahabi, and Li Xiong. 2018. Privacy-preserving online task assignment in spatial crowdsourcing with untrusted server. In *ICDE*.
[109] Yongxin Tong, Lei Chen, and Cyrus Shahabi. 2017. Spatial Crowdsourcing: Challenges, Techniques, and Applications. *VLDB Endowment [Tutorial]* 10, 12 (2017), 1988–1991.
[110] Yongxin Tong, Jieying She, Bolin Ding, Lei Chen, Tianyu Wo, and Ke Xu. 2016. Online minimum matching in real-time spatial data: experiments and analysis. *VLDB* 9, 12 (2016), 1053–1064.
[111] Yongxin Tong, Jieying She, Bolin Ding, Libin Wang, and Lei Chen. 2016. Online mobile micro-task allocation in spatial crowdsourcing. In *ICDE*. 49–60.
[112] Yongxin Tong, Libin Wang, Zimu Zhou, Bolin Ding, Lei Chen, Jieping Ye, and Ke Xu. 2017. Flexible online task assignment in real-time spatial data. *VLDB Endowment* 10, 11 (2017), 1334–1345.

[113] Luan Tran, Hien To, Liyue Fan, and Cyrus Shahabi. 2017. A Real-Time Framework for Task Assignment in Hyperlocal Spatial Crowdsourcing. *ACM Transactions on Intelligent Systems and Technology* (2017), 37.

[114] Umair ul Hassan and Edward Curry. [n. d.]. Flag-verify-fix: adaptive spatial crowdsourcing leveraging location-based social networks. In *GIS*.

[115] Heli Väätäjä, Teija Vainio, and Esa Sirkkunen. 2012. Location-based crowdsourcing of hyperlocal news: dimensions of participation preferences. In *GROUP*. 85–94.

[116] Bregtje Van der Haak, Michael Parks, and Manuel Castells. 2012. The future of journalism: Networked journalism. *Int. Journal of Communication* 6 (2012), 16.

[117] Maria Vasardani, Stephan Winter, Kai-Florian Richter, Lesley Stirling, and Daniela Richter. 2012. Spatial interpretations of preposition "at". In *GIS GEOCROWD*. 46–53.

[118] Gang Wang, Bolun Wang, Tianyi Wang, Ana Nika, Haitao Zheng, and Ben Y Zhao. 2016. Defending against sybil devices in crowdsourced mapping services. In *MobiSys*. 179–191.

[119] Liang Wang, Zhiwen Yu, Qi Han, Bin Guo, and Haoyi Xiong. 2018. Multi-objective Optimization based Allocation of Heterogeneous Spatial Crowdsourcing Tasks. *IEEE Transactions on Mobile Computing* (2018), 1637–1650.

[120] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*. 2035–2043.

[121] Raymond Chi-Wing Wong, Yufei Tao, Ada Wai-Chee Fu, and Xiaokui Xiao. 2007. On efficient spatial matching. In *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, 579–590.

[122] Mingjun Xiao, Jie Wu, Liusheng Huang, Yunsheng Wang, and Cong Liu. 2015. Multi-task assignment for crowdsensing in mobile social networks. In *INFOCOM*. 2227–2235.

[123] Mingjun Xiao, Jie Wu, Sheng Zhang, and Jiapeng Yu. 2017. Secret-sharing-based secure user recruitment protocol for mobile crowdsensing. In *INFOCOM*.

[124] Xiaojuan Xie, Haining Chen, and Hongyi Wu. 2009. Bargain-based stimulation mechanism for selfish mobile nodes in participatory sensing network. In *SECON*. 1–9.

[125] Tingxin Yan, Matt Marzilli, Ryan Holmes, Deepak Ganesan, and Mark Corner. 2009. mCrowd: a platform for mobile crowdsourcing. In *SenSys*. 347–348.

[126] Dejun Yang, Guoliang Xue, Xi Fang, and Jian Tang. 2012. Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing. In *MobiCom*. 173–184.

[127] Peng Yang, Ning Zhang, Shan Zhang, Kan Yang, Li Yu, and Xuemin Shen. 2017. Identifying the Most Valuable Workers in Fog-Assisted Spatial Crowdsourcing. *IEEE Internet of Things Journal* 4, 5 (2017), 1193–1203.

[128] Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. 2011. A survey of crowdsourcing systems. In *SocialCom*. 766–773.

[129] Yuxiang Zeng, Yongxin Tong, Lei Chen, and Zimu Zhou. 2018. Latency-oriented task completion via spatial crowdsourcing. In *ICDE*.

[130] Bo Zhang, Chi Harold Liu, Jianyu Lu, Zheng Song, Ziyu Ren, Jian Ma, and Wendong Wang. 2016. Privacy-Preserving QoI-Aware Participant Coordination for Mobile Crowdsourcing. *Computer Networks* (2016), 29–41.

[131] Lefeng Zhang, Xiaodan Lu, Ping Xiong, and Tianqing Zhu. 2015. A differentially private method for reward-based spatial crowdsourcing. In *ATIS*. 153–164.

[132] Xinglin Zhang, Zheng Yang, Yue-Jiao Gong, Yunhao Liu, and Shaohua Tang. 2017. SpatialRecruiter: maximizing sensing coverage in selecting Workers for Spatial Crowdsourcing. *IEEE Transactions on Vehicular Technology* 66, 6 (2017), 5229–5240.

[133] Xinglin Zhang, Zheng Yang, Yunhao Liu, and Shaohua Tang. 2016. On Reliable Task Assignment for Spatial Crowdsourcing. *IEEE Transactions on Emerging Topics in Computing* (2016), 1–1.

[134] Yongjian Zhao and Qi Han. 2016. Spatial crowdsourcing: current state and future directions. *IEEE Communications Magazine* 54, 7 (2016), 102–107.

[135] Yan Zhao, Yang Li, Yu Wang, Han Su, and Kai Zheng. 2017. Destination-aware Task Assignment in Spatial Crowdsourcing. In *CIKM*. 297–306.

[136] Liu Zheng and Lei Chen. 2016. Mutual benefit aware task assignment in a bipartite labor market. In *ICDE*. 73–84.

[137] Gaoqiang Zhuo, Qi Jia, Linke Guo, Ming Li, and Pan Li. 2017. Privacy-preserving verifiable set operation in big data for cloud-assisted mobile crowdsourcing. *IEEE Internet of Things Journal* 4, 2 (2017), 572–582.

[138] Matthew Zook, Mark Graham, Taylor Shelton, and Sean Gorman. 2010. Volunteered geographic information and crowdsourcing disaster relief: a case study of the Haitian earthquake. *World Medical & Health Policy* 2, 2 (2010), 7–33.

Received March 2018; revised 1; accepted 1