Expert Systems With Applications 58 (2016) 36–56

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Efficient task assignment for spatial crowdsourcing: A combinatorial fractional optimization approach with semi-bandit learning

Umair ul Hassan*, Edward Curry

The Insight Centre for Data Analytics, National University of Ireland Galway, The DERI Building, IDA Business Park, Lower Dangan, Galway, Ireland

ARTICLE INFO

Article history:
Received 29 June 2015
Revised 10 March 2016
Accepted 10 March 2016
Available online 1 April 2016

Keywords:
Spatial crowdsourcing
Task assignment
Combinatorial fractional programming
Multi-armed bandit

ABSTRACT

Spatial crowdsourcing has emerged as a new paradigm for solving problems in the physical world with the help of human workers. A major challenge in spatial crowdsourcing is to assign reliable workers to nearby tasks. The goal of such task assignment process is to maximize the task completion in the face of uncertainty. This process is further complicated when tasks arrivals are dynamic and worker reliability is unknown. Recent research proposals have tried to address the challenge of dynamic task assignment. Yet the majority of the proposals do not consider the dynamism of tasks and workers. They also make the unrealistic assumptions of known deterministic or probabilistic workers' reliabilities. In this paper, we propose a novel approach for dynamic task assignment in spatial crowdsourcing. The proposed approach combines bi-objective optimization with combinatorial multi-armed bandits. We formulate an online optimization problem to maximize task reliability and minimize travel costs in spatial crowdsourcing. We propose the *distance-reliability ratio* (DRR) algorithm based on a combinatorial fractional programming approach. The DRR algorithm reduces travel costs by 80% while maximizing reliability when compared to existing algorithms. We extend the DRR algorithm for the scenario when worker reliabilities are unknown. We propose a novel algorithm (DRR-UCB) that uses an *interval estimation* heuristic to approximate worker reliabilities. Experimental results demonstrate that the DRR-UCB achieves high reliability in the face of uncertainty. The proposed approach is particularly suited for real-life dynamic spatial crowdsourcing scenarios. This approach is generalizable to the similar problems in other areas in expert systems. First, it encompasses online assignment problems when the objective function is a ratio of two linear functions. Second, it considers situations when intelligent and repeated assignment decisions are needed under uncertainty.

1. Introduction

Crowdsourcing systems have emerged as a new form of multi-agent systems where three types of agents interact with each other, namely requesters, workers, and platform. The requesters submit tasks on platform that are performed by workers. The platform serves as a mediator that provides appropriate assignment of tasks to workers to maximize the utility of crowd work (Deng, Shahabi, & Demiryurek, 2013; Hassan & Curry, 2014). Crowdsourcing has been applied in variety domains, such as machine learning (Chen, Lin, & Zhou, 2013b), natural language processing (Tarasov, Delany, & Mac Namee, 2014), and mobile-based sensing (Ganti, Ye, & Lei, 2011). Spatial crowdsourcing is a form of crowdsourcing that employs crowd workers for performing tasks in the physical world at various locations (Cheng et al., 2015; Kazemi and Shahabi, 2012; To, Ghinita, and Shahabi, 2014; To, Shahabi, & Kazemi, 2015). The use of spatial crowdsourcing is further illustrated with the help of the following scenario:

Consider the situation, where a requester is interested in collecting high quality and representative photos of disaster hit areas in a country. The locations of interest are spread across the country. The requester designs a task for each location i.e. spatial crowdsourcing task. The requester is interested in the coverage of all locations with high quality results. Fig. 1 illustrates such a scenario on a map.

Most of the existing crowdsourcing platforms serve as a marketplace with rudimentary task assignment functionality, such as Amazon Mechanical Turk, TaskRabbit, and ClickWorker (Horton & Chilton, 2010; Musthag & Ganesan, 2013). In fact, these platforms largely rely on workers to assign tasks to themselves when visiting the platform. Tasks may not be assigned to appropriate workers in this manual approach, also known as *worker selected tasks*

* Corresponding author. Tel.: +353 85 729 8904.
E-mail addresses: umair.ulhassan@insight-centre.org (U. ul Hassan), edward.curry@insight-centre.org (E. Curry).

http://dx.doi.org/10.1016/j.eswa.2016.03.022

Fig. 1. Example of spatial crowdsourcing on the map of Haiti after the 2010 earth quake. A new spatial task (in blue) requests recent photos of a building at the indicated location.

(WST), due to search friction issues (Kittur et al., 2013; Kulkarni et al., 2012). Recent research has focused on developing algorithmic approaches to task assignment with the aim of addressing the limitations of the WST approach. The algorithmic approach, also known as the *server assigned tasks* (SAT) approach, formulates the dynamic task assignment as a sequential decision making problem. The sequential decision making involves matching dynamically arriving tasks with dynamically arriving workers over time (Abraham, Alonso, Kandylas, and Slivkins, 2013; Hassan and Curry, 2014; Ho, Jabbari, & Vaughan, 2013).

In this paper, we focus our attention to the SAT-based dynamic task assignment problem in spatial crowdsourcing. Specifically, we introduce and formulate the *minimum-cost maximum reliability assignment* (MC-MRA) problem in spatial crowdsourcing. The MC-MRA problem aims to maximize reliability and minimize the travel costs of spatial tasks. Existing literature on spatial crowdsourcing generally assumes deterministic settings for task assignment i.e. each assignment is assumed to result in task completion with high quality (Deng et al., 2013; Kazemi and Shahabi, 2012; To et al., 2014, 2015). By comparison, the MC-MRA problem considers stochastic settings where the reliability of an assignment is defined in terms of the probability that spatial task will be completed with high quality by the assigned worker (Cheng et al., 2015). Assuming that the reliabilities of different assignment choices are known, we reduce the MC-MRA problem to *minimum-cost maximum weight bipartite matching* problem and adapt two existing approaches to address the reduced problem (To et al., 2015). Our experimental evaluation shows that the adapted approaches do not jointly optimize reliability and travel costs; therefore, resulting higher travel costs under various conditions. To address the limitations of existing approaches, we propose a novel approach based on *combinatorial fractional programming*. The proposed approach aims to dynamically assign tasks such that both the reliability of assignments is maximized and the travel costs are minimized, for all spatial tasks.

Further, we relax the assumption of known reliabilities and consider the MC-MRA problem with online learning. Online learning necessitates the dynamic estimation of worker reliabilities based on the observed outcomes of task assignments. A fundamental challenge of dynamic assignment with estimated reliabilities is to address the dilemma of learning versus optimization, also known as the *exploration-exploitation* trade-off (Barto, 1998) in literature. *Exploration* involves choosing workers for the purpose of learning their reliability. *Exploitation* entails using the gained knowledge to optimize the assignment objective. Existing research works have employed primal-dual (Ho et al., 2013) and multi-armed bandit (Abraham et al., 2013; Chen et al., 2013b; Tran-Thanh, Stein, Rogers, & Jennings, 2014) techniques to address the trade-off. These techniques have primarily focused on the non-spatial crowdsourcing scenarios and use simple cost constraints (Slivkins & Vaughan, 2013). Instead, we address the exploration-exploitation trade-off for the MC-MRA problem that optimizes both the reliability and the travel costs in spatial crowdsourcing. The specific research contributions of this article are summarized below:

- We introduce and formalize the MC-MRA problem based on the SAT-based spatial crowdsourcing. We reduce the MC-MRA problem to the *minimum-cost maximum weight bipartite matching* problem and adapt two existing approaches to address the reduced problem.
- We propose a novel *distance-reliability ratio* (DRR) approach for the MC-MRA problem, that is based on combinatorial fractional programming. The DRR approach employs Newton's method to transform the fractional assignment problem to an equivalent parameterized linear assignment problem.
- We extend the DRR approach, for adaptive assignment, to enable the estimation of worker reliabilities from observed outcomes of previous task assignments. We propose two adaptive DRR algorithms based on combinatorial multi-armed bandit model with semi-bandit learning. The DRR-GRD is inspired by the *greedy exploration* approach and the and DRR-UCB algorithm follow an *interval estimation* approach.
- We extensively evaluate the performance of proposed algorithms against adapted algorithms on synthetic and real-world datasets. The performance results establish the effectiveness of the DRR algorithm and its variants in terms of reliability and travel costs. The results also establish the effectiveness of adaptive DRR algorithms in terms of estimating worker reliabilities.

This paper extends the research on experts systems in two key ways. First, it considers the online algorithms that aim to optimize a bi-objective objective function. One set of optimization variables are probabilistic and the second are deterministic. Second, it combines semi-bandit learning with the bi-objective optimization. Semi-bandit learning assumes that the probabilistic variables are unknown and must be approximated by observing the outcomes of optimization decisions.

The rest of this article is organized as follows. In Section 2, we summarize the related research on the assignment problem in spatial crowdsourcing. We highlight the research gaps in existing literature on spatial crowdsourcing and its related topics. Section 3 provides the necessary definitions of concepts in SAT-based spatial crowdsourcing and Section 3.2 describes the basic *maximum reliability assignment* (MRA) problem. In Section 4, we introduce the *minimum-cost maximum reliability assignment* (MC-MRA) problem and present our proposed DRR approach for efficient solution to the MC-MRA problem. Section 5 extends the proposed DRR approach with online learning based algorithms. We evaluate our proposed algorithms using real-world and synthetic datasets in Section 6. Section 7 discusses the implications of the proposed approaches and their performance results. We conclude the paper in Section 8 and layout plans for future work.

2. Related work

Spatial crowdsourcing is distinguished from other forms of crowdsourcing by the fact that workers are required to visit locations in the physical world to perform tasks. One primary challenge of spatial crowdsourcing is matching tasks with appropriate workers on the ground (Kazemi and Shahabi, 2012; To et al., 2014, 2015). Kazemi and Shahabi proposed a taxonomy of spatial crowdsourcing that highlights two modes of task assignment:

worker selected tasks (WST) and *server assigned tasks* (SAT). The majority of commercial crowdsourcing platforms employ WST for task assignment, which has also been used in spatial crowdsourcing (Chen et al., 2014; Deng et al., 2013; Thebault-Spieker, Terveen, & Hecht, 2015). The WST method emphasizes self determination of tasks to be performed by a worker, i.e. workers explicitly visits the crowdsourcing platform and self-assign tasks through an appropriate search and browse interface. The WST method is prone to search friction issues, when workers have difficultly in finding the right tasks, or vice versa (Kittur et al., 2013). Search friction can arise due to the limitations of the interaction mechanism and the user interface design (Kulkarni et al., 2012). The SAT method addresses this issue by algorithmically managing the assignment process (Hassan & Curry, 2013; 2014; Kazemi & Shahabi, 2012; To et al., 2014). The SAT method relies upon knowledge about tasks and workers to find suitable matches. Besides task assignment, the Kazemi and Shahabi taxonomy also describes the dimensions of incentives and redundancy for spatial crowdsourcing. The focus of this paper is SAT-based dynamic tasks assignment for self-incentivized crowdsourcing.

Kazemi and Shahabi also proposed and formulated the *maximum task assignment* (MTA) problem for spatial crowdsourcing. They further proposed a network-flow based algorithm for addressing the MTA problem; however, their proposal is based on the assumption of deterministic outcomes of assignments. The MTA problem was extend to the *maximum score assignment* (MSA) problem for skills-based spatial crowdsourcing, also under deterministic settings (To et al., 2015). To et al. defined a privacy enabling framework for dynamic assignment in spatial crowdsourcing (To et al., 2014). The framework is designed to hide the actual locations of workers during the assignment process while maximizing the number of assigned tasks. Deng et al. proposed approximation algorithms for scheduling tasks for WST-based spatial crowdsourcing (Deng et al., 2013). The proposed algorithms aim to maximize the number of tasks performed by an individual worker. All of the above mentioned works consider the deterministic settings for assignment outcomes; by comparison, this paper assume more challenging stochastic settings. Cheng et al. recently introduced the diversity-based spatial crowdsourcing with probabilistic reliabilities; however, their proposed problem does not optimize travel costs and assumes known worker reliabilities. Hassan and Curry formalized the dynamic assignment problem for spatial crowdsourcing, under stochastic settings with online learning (Hassan & Curry, 2014). Their formalization is limited to single assignment decisions without considering travel costs; on the other hand, this paper considers a more generalized combinatorial assignment problem with multi-objective optimization with online learning.

Mobile crowdsensing is another class of crowdsourcing that focuses on exploiting the sensors on mobile devices for collection of data in physical environments (Zhang, Wang, Xiong, & Guo, 2014). Mobile crowdsensing is differentiated from spatial crowdsourcing due to the focus on the use of mobile sensors, sometimes without the need of explicit user interaction (Zhang et al., 2014). The *optimal task allocation* (OTA) problem under time constraints has been addressed recently (Feng, Zhu, Zhang, Ni, & Vasilakos, 2014). Xiao, Wu, Huang, Wang, and Liu proposed a multi-task assignment approach for workers from mobile social networks (Xiao et al., 2015). Other approaches include assigning tasks based on worker trajectory, also known as the *task orienteering* problem (Chen et al., 2014; Chen, Cheng, Lau, & Misra, 2015). These research works also do not consider multi-objective optimization and adaptivity for dynamic assignment in mobile crowdsensing. Therefore, the techniques discussed in this paper are more complimentary to these works. A more detail analysis of related literature if provided in Appendix A.

3. Preliminaries

We first define a set of preliminaries in the context of dynamic task assignment under stochastic settings, for self-incentivized SAT-based spatial crowdsourcing.

Definition 1 (Spatial task)**.** A spatial task t at time instance r_{st} requires workers to travel to at location l_t and perform a predefined set of actions or collect information. The task must be performed by at least one worker; furthermore, each task is set to expire after time instance r_{et}.

Note that each spatial task requires a worker to physically travel to the task location. For example, a task might ask the worker to take photos of an event for the purpose of documenting that event. After a task becomes visible in the crowdsourcing system, it must be finished before expiry. Otherwise the task is considered incomplete which results in a lower than desired converge of the associated event. Task are submitted dynamically to the crowdsourcing system and disappear from it after completion or expiry.

Definition 2 (Mobile worker)**.** A worker w is a person who volunteers to perform spatial tasks by sending requests for tasks, to the spatial crowdsourcing platform. When a worker is ready to perform a task she sends the task request along with her current location l_w. Since the worker is moving around her location dynamically changes over time.

A worker indicants her availability through a task request. Previous research works have considered the case when workers state their task capacity and spatial constraints along with the task requests (Kazemi & Shahabi, 2012). In reality, the requirement to specify such constraints is burdensome requiring additional decision making from workers. We relax this requirement by considering the situation where each worker performs on one task at a time and requests the next task after completing the current task. Note that, a worker might have the spatial preferences in terms of the maximum distance she is willing to travel. We assume that such preferences are private to the worker and the crowdsourcing platform does not necessarily require such preference with the task requests.

Given the spatial tasks and worker requests a SAT-based platform performs the assignment and gathers results. The interaction between requesters, platform, and workers is a repeating dynamic process. Understandably, task assignment is also not a once-off process as tasks and worker arrive dynamically on the platform. The assignment process proceeds in rounds where each round involves matching tasks with workers.

Definition 3 (Assignment round)**.** In round r, let $T_r = \{t_1, \dots, t_n\}$ be the set of incomplete tasks and $W_r = \{w_1, \dots, w_m\}$ be the set of available workers. An assignment is defined as a pair $x_{i,j} = < t_i, w_i >$ when task $t_i \in T_r$ is assigned to worker $w_j \in W_r$. Let $X_r \subset \mathcal{X}_r$ denote the set of assignments in the current round, where $\mathcal{X}_r$ is the set of all possible assignments.

The set $\mathcal{X}_r$ must satisfy the constraints that each worker is assigned at most one task in a round and vice versa. A new worker is assigned to the task in a round until the task is removed from the task list T_r, due to successful completion or expiry. Note that, a list of previous assignments is maintained for each task to prevent the same worker being assigned to the same task multiple times. The task assignment set chosen by a particular assignment strategy π is denoted by X_r^{π}. The goal of an online assignment algorithm is to find the best strategies for all rounds $\{X_1, X_2, \dots\}$ under uncertainty. The uncertainty arises due to the fact that a worker might not complete the assigned task by the end of a round. At the end of a round, the outcomes of all assignments are observed. The outcome of an assignment $x_{i,j}$ is represented by variable $y_{i,j}$. If a task is successfully completed then $y_{i,j} = 1$ and 0 otherwise. We define the notion of worker reliability as the likelihood that a worker will complete an assigned task.

Definition 4 (Worker reliability)**.** Each worker w_j has a reliability $p_{i,j}$ for task t_i. If worker w_j is assigned to task t_j, then the worker completes the task before the end of the round with probability $p_{i,j}$, i.e.:

$$y_{i,j} = \begin{cases} 1 & \text{with probability } p_{i,j} \\ 0 & \text{otherwise} \end{cases}$$

Without loss of generality we assume that $p_{i,j} \in (0, 1)$. The reliability of worker w_j on task t_i depends on task, worker, and their context. Let $\mathcal{W}$ be the set of all workers known to the crowdsourcing platform. Each worker $w_j \in \mathcal{W}$ has an associated average reliability q_j that is defined as the expectation of reliabilities of assignments to the worker. Essentially the reliabilities $\{p_{1,j}, p_{2,j}, ...\}$ of assignments to the worker w_j are sampled from a bounded distribution with mean q_j. First we assume that the worker reliabilities are known during a round; however, we relax this assumption later on in Section 5.

Definition 5 (Task reliability)**.** Let W_t be the set of workers assigned to a task t before it is completed or expires. Then the probability that atleast one worker completes the task is defined as the reliability of t.

The objective of repeated assignment over multiple rounds is to assign workers to a task in such a way that the chance of task completion is maximized in each assignment. Therefore, in each round the assignment algorithm may choose assignments such that their combined reliabilities are maximized.

Definition 6 (Travel costs)**.** Each assignment $x_{i,j}$ has an associated distance $d_{i,j}$ that quantifies the cost in terms of travel required from worker w_j while assigned to the task t_i.

The distance between task and worker location plays a significant role in spatial crowdsourcing. It is intuitive to assume that workers would find it easier to perform tasks near to them. From a requester's perspective the worker in the near vicinity of a task has a higher chance of completing the task in time. The distance may be calculated according to different metrics: Euclidean distance, haversine formula, or time. We assume that one consistent metric is used to quantify distance between two locations; therefore, smaller distances are preferred by both worker and requesters.

3.1. Assignment protocol

The assignment process in spatial crowdsourcing platform proceeds in discrete rounds $r = \{1, 2, ..., \infty\}$, where each round spans for τ amount of time. A round consists of following steps:

1. A combined set of newly submitted and previously incomplete tasks T_r is observed, where each task has its associated location.
2. A set of available workers W_r is observed, where each worker has her associate location.
3. A set of worker reliabilities $\{p_{i,j} | t_i \in T_r, w_j \in W_r\}$ are observed for all possible assignments.
4. The assignment algorithm selects the set of assignments $X_r \in \mathcal{X}_r$ and assigns tasks to appropriate workers.
5. At the end of a round (when τ duration has elapsed), the platform observes the tasks completed with high quality and updates the task list T_{r+1}.

Note that a task is added to T_r as soon as it is submitted by the requester and removed after completion or expiry. Here we assume that the duration between task submission and task expiry is in multiple of rounds. Fig. 2 illustrates the assignment protocol in detail for three example rounds. In first the round, there are three tasks (circles) and three workers (squares). Each task is assigned to a worker; however, some workers complete tasks (shown by circles without any color) and some workers do not (shown by colored circles). In the second round, a new task appears into the system and another worker becomes available. An incomplete task is assigned to another worker while ensuring that previously assigned workers are not assigned to the same task again (as highlighted by dotted lines). The process continues to the next round, as completed tasks disappear from the system. While we assume that each task requires one worker, it is straight forward to consider the task that requires multiple workers as multiple instances of the same task require a worker.

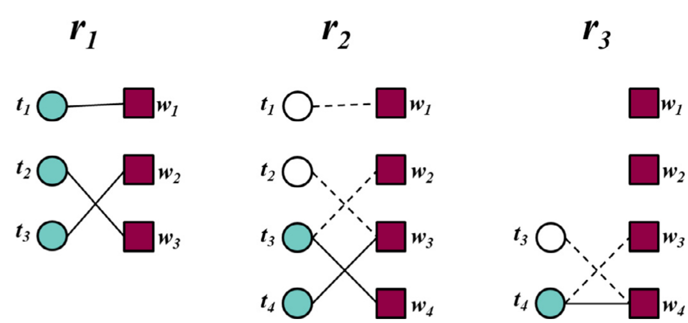

Fig. 2. Example of three rounds of the task assignment protocol. Circles and squares represent tasks and workers respectively. An empty circle represents a task completed in the previous round. A line represents an assignment in the current round and the dotted lines show assignments from previous rounds to a task

3.2. Maximum reliability assignment

We now introduce the *maximum reliability assignment* (MRA) problem for spatial crowdsourcing. Then we describe the MRA problem and reduce it to the *maximum weight bipartite matching* (MWMB) problem to solve it efficiently.

Definition 7 (Maximum reliability assignment)**.** Given a time interval $\phi = \{r_1, r_2, ..., r_z\}$, let $R(X_r)$ be the reliability of assignment set X_r in round r, i.e.:

$$R(X_r) = \prod_{(i,j) \in X_r} p_{i,j} \tag{1}$$

The problem of maximum reliability assignment is to assign workers to tasks such that the total reliability is maximized over all rounds in ϕ, i.e.:

$$Rel_{\phi} = \max \prod_{r \in \phi} R(X_r) \tag{2}$$

A global solution to the MRA problem is not feasible unless the assignment process is clairvoyant (i.e. all information about rounds and reliabilities is known beforehand). The MRA problem requires design of an online algorithm to which the sets T_r and W_r are revealed at the start of each round. An offline version of the MRA problem would mean that number of rounds, task arrival time, task expiry, and worker reliabilities are known a priori. The offline version of the MRA problem is in indeed an NP-Hard problem by reduction to the *generalized assignment problem* (GAP) (Pentico, 2007). If the start and end time of all tasks are same and all workers are available through all rounds, the offline MRA problem is equivalent to the GAP. Therefore, the GAP is a special case of the offline MRA problem. Since GAP is an NP-Hard problem, the offline MRA is also an NP-Hard problem. To address the MRA problem, the platform aims to maximize reliability of assignment set $R(X_r)$

locally in each round. Therefore, the objective of a local optimization strategy is to exploit information limited to the current round.

3.3. Maximum weighted bipartite matching

We introduce a location optimization strategy that reduces the local MRA problem to the *maximum weight bipartite matching* (MWBM) problem. In a local optimization strategy for round r, we have a set of available worker W_r and set of incomplete tasks T_r and the goal is find an assignment set X_r that maximizes $R(X_r)$. We rewrite the reliability of an assignment set (see eq. 1) as the reliability score:

$$RS(X_r) = \ln(R(X_r)) = -\ln\left(\prod_{(i,j)\in X_r} p_{i,j}\right) = \sum_{(i,j)\in X_r} -\ln(p_{i,j}) \tag{3}$$

Based on above equation, the goal of maximizing the reliability for all rounds (see Eq. (2)) is equivalent to maximizing $RS(X_r)$ for all rounds. Given the time interval ϕ, the global objective is to maximize the sum of assignment scores for all rounds, i.e. $\Sigma_{r\in\phi} RS(X_r)$.

Our goal is to choose an assignment set X_r such that the overall reliability score is maximized; thereby, solving the local MRA problem.

Theorem 1. *The maximum reliability assignment problem is reducible to the maximum weighted bipartite matching problem.*

Proof. We prove the theorem for a round r in which the set of available workers is W_r and the set of incomplete tasks is T_r. Let $G_R = (V, E)$ be an undirected bipartite graph. The set of vertices $V_r = T_r \cup W_r$ is partitioned; such that, each task $t_i \in T_r$ represents a vertex on the left side and each worker $w_j \in W_r$ represents a vertex on the right size (as shown in Fig. 2). The set of edges $e_{i,j} \in E_r$ connect vertices in T_r with vertices in W_r. A match $< t_i, w_j >$ is deemed valid only if the vertex for task t_i and the vertex for worker w_j appear in at most one edge in E. Meaning that each task is assigned at most one worker and each worker gets at most one task. We associate a weight with each edge $e_{i,j}$ according to following reliability score:

$$score(e_{i,j}) = \begin{cases} 0 & \text{if } e_{i,j} \in F_r \\ -\ln(p_{i,j}) & \text{otherwise} \end{cases} \tag{4}$$

where F_r is list of unsuccessful assignments for the tasks not yet complete. The zero score discourages the assignment of the same worker to an incomplete task. Subsequently, the local MRA problem reduces to finding the maximal weight bipartite matching in graph G_r. □

Based on the above reduction, we can use existing algorithms for the MWBM problem to find a solution for the local MRA problem. The MWBM problem can be solved in polynomial time using algorithms developed for network flow problems (Ahuja, Magnanti, & Orlin, 1993) or linear sum assignment problems (Burkard, Dell'Amico, & Martello, 2009). In this paper, we employ the well-known Hungarian algorithm for solving the MWBM problem in each round. Let $n = |T_r|$ and $m = |W_r|$, we generate a cost matrix C_r based on edge weights of the MWBM problem . If *scoreMAX* is the maximum weight among all edges, then the cost of edge $e_{i,j}$ is set to $c_{i,j} = scoreMAX - score(e_{i,j})$ The following integer linear program specifies the assignment problem to be solved using the Hungarian algorithm:

$$\begin{aligned} &\min \sum_{i=1}^{n}\sum_{j=1}^{m} c_{i,j}\cdot x_{i,j} \\ &\text{s.t.} \sum_{j=1}^{m} x_{i,j} = 1 \qquad \forall i \\ &\sum_{i=1}^{n} x_{i,j} = 1 \qquad \forall j \\ &x_{i,j} \in \{0,1\} \qquad \forall (i,j) \end{aligned} \tag{5}$$

where the binary variable $x_{i,j}$ indicates assignment of a task to a worker. We use the Jonker–Volegenant variant of the Hungarian algorithm which has time complexity of $O(n^3)$, where n is the largest dimension of cost matrix (Jonker & Volgenant, 1987). Eq. (5) considers one task per worker constraint in a round. This constraint can be relaxed when workers are willing to perform more than one task. Let B_j be the number of tasks that the worker w_j is willing to perform in a round. In this case, worker w_j is represented with B_j nodes in right hand partition of graph G_r to utilize Hungarian algorithm. In this case, each worker is assigned B_j nearest tasks.

The MWBM approach discussed here optimizes reliability locally for each round; therefore, the global optimization is not guaranteed. Furthermore, this approach only considers the reliability as the optimization criteria while ignoring spatial characteristics of tasks and workers. Such a naive approach may result is unnecessary burden on worker in terms of travel to the task location. To improve over this approach, we propose the *minimum-cost maximum reliability assignment* (MC-MRA) problem that also optimizes the distance between tasks and workers.

4. Minimum-cost maximum reliability assignment

In Section 3, we defined the assignment distance between a task t_i and worker w_j. We associate the travel cost with every pair of task and worker by calculating the distance between them. The basic idea is to incorporate the travel cost in the assignment process. Subsequently, the optimization is not only targeted at the reliability but spatial characteristics as well. The goal of MC-MRA problem is to find task and worker matches such that the reliability is maximized and travel costs are minimized. In the following theorem, we reduce the local MC-MRA problem to a minimum-cost MWBM problem.

Theorem 2. *The local minimum-cost maximum reliability assignment problem is reducible to the minimum-cost maximum weight bipartite matching problem.*

Proof. Similar to the Theorem 1, our proof is based on a round r where assignments are made over a set of tasks T_r and a set of workers W_r. Let $G'_r = (V, E)$ be the undirected weighted bipartite graph constructed in the similar way as in proof of Theorem 1. Each $e_{i,j}$, in the G_r, has an associated weight $score(e_{i,j})$. We associate a cost $distance(e_{i,j})$ with each edge based on the distance between the locations of task and worker. Each edge in G'_r has associated weight and cost (i.e. reliability score and travel cost, respectively). Subsequently, the solution to local minimum-cost MRA problem reduces to finding a bipartite matching with minimum-cast and maximum weight in G'_r □

4.1. Close distance priority

A simple approach is to solve each optimization criteria of the MC-MRA problem sequentially (Pentico, 2007). The first step is to find assignment sets with the maximize possible value for the reliability objective. Then the assignment set with minimum travel costs is selected among those available maximal choices. This approach is also known as the *close distance priority* approaches (To et al., 2015). For each round, the Hungarian algorithm is used to find the maximal score assignment set X_r^{MAX} using Eq. (5). Let f_{max} be the total reliability score of the maximal score assignment set, i.e.:

$$f_{max} = \sum_{(i,j)\in X_r^{MAX}} score(e_{i,j})$$

Then the travel cost minimization problem is reformulated as the following integer program:

$$\begin{aligned} \min \sum_{i=1}^{n}\sum_{j=1}^{m} d_{i,j}\cdot x_{i,j} & \\ \text{s.t.} \sum_{i=1}^{n}\sum_{j=1}^{m} s_{i,j}\cdot x_{i,j} \geq fmax & \\ \sum_{j=1}^{m} x_{i,j} = 1 \qquad & \forall i \\ \sum_{i=1}^{n} x_{i,j} = 1 \qquad & \forall j \\ x_{i,j} \in \{0,1\} \qquad & \forall (i,j) \end{aligned} \tag{6}$$

where $d_{i,j} = distance(e_{i,j})$ and $s_{i,j} = score(e_{i,j})$. We employ a branch and cut technique for solving the cost minimization problem in Eq. (6). The resulting assignment set X_r is the minimal cost assignment set among all feasible assignment sets with the total reliability score of at least f_{max}. Similar to the MWBM approach the CDP approach finds solution for the local instance of the MC-MRA problem in each round. The computational complexity of the CDP approach is dominated by the branch and cut algorithm used to solve the integer program.

4.2. Distance-reliability ratio

The CDP approach is applicable to situations when there are more than one feasible solution for the MRA problem; therefore, priority is given to the solution with smallest total travel costs. This approach does not always results in cost minimization. Specifically in situations when the maximal solutions found in first round also incur high travel costs. To address this issue, we propose the *distance-reliability ratio* (DRR) approach based on a joint optimization of the reliability score and travel costs. We re-formulate the minimum-cost MWBM problem as described in Theorem 2. Our formulation is based on the *combinatorial fractional programming* (Bajalinov, 2003), as represented by the following integer linear-fractional program:

$$\begin{aligned} \min \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} d_{i,j}\cdot x_{i,j}}{\sum_{i=1}^{n}\sum_{j=1}^{m} s_{i,j}\cdot x_{i,j}} & \\ \text{s.t.} \sum_{j=1}^{m} x_{i,j} = 1 \qquad & \forall i \\ \sum_{i=1}^{n} x_{i,j} = 1 \qquad & \forall j \\ x_{i,j} \in \{0,1\} \qquad & \forall (i,j) \end{aligned} \tag{7}$$

Finding a direct solution to above nonlinear program is a difficult problem. Dinkelbach proposed a parametric approach by iteratively solving an equivalent linearized version of linear-fractional programs, also known as Newton's method (Dinkelbach, 1967). We employ this method to transform the Eq. (7) to an equivalent integer linear program using Charnes and Cooper transformation (Charnes & Cooper, 1962). The following integer program formulates the transformed integer linear program with parameter λ:

$$\begin{aligned} \min \sum_{i=1}^{n}\sum_{j=1}^{m} (d_{i,j} - \lambda s_{i,j})\cdot x_{i,j} & \\ \text{s.t.} \sum_{j=1}^{m} x_{i,j} = 1 \qquad & \forall i \\ \sum_{i=1}^{n} x_{i,j} = 1 \qquad & \forall j \\ x_{i,j} \in \{0,1\} \qquad & \forall (i,j) \end{aligned} \tag{8}$$

Similar to the approaches discussed previously, we employ the Hungarian algorithm to solve the transformed assignment problem (Megiddo, 1979). To discourage assignment of previously unsuccessful workers to an incomplete task, we replace the cost coefficient $(d_{i,j} - \lambda s_{i,j})$ with a reasonably high value. The parameter λ is calculated using the Newton's method in each iteration. Algorithm 1 summarizes the parametric algorithm for solving MC-MRA using the DRR approach. The DRR algorithm requires an optimality parameter δ which takes on reasonably small values. Our proposed approach quickly coverages to a solution; since, the number of iterations for the Newton's method have strong polynomial bounds (Megiddo, 1979; Radzik, 1992).

Algorithm 1 The DRR algorithm

Require: $\delta, [s_{i,j}]^{n\times m}, [d_{i,j}]^{n\times m}$
1: $[x_{i,j}]^{n\times m} \leftarrow [0]^{n\times m}$ {Arbitrary assignment}
2: **repeat**
3:
$$\lambda \leftarrow \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} d_{i,j}\cdot x_{i,j}}{\sum_{i=1}^{n}\sum_{j=1}^{m} s_{i,j}\cdot x_{i,j}}$$
4: $[c_{i,j}]^{n\times m} \leftarrow [d_{i,j}]^{n\times m} - \lambda \cdot [s_{i,j}]^{n\times m}$
5: $[x_{i,j}]^{n\times m} \leftarrow \text{Hugarian}([c_{i,j}]^{n\times m})$
6: $f_\lambda \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{m} (d_{i,j} - \lambda s_{i,j})\cdot x_{i,j}$
7: **until** $f_\lambda \geq \delta$
8: **return** $[x_{i,j}]^{n\times m}$

5. Dynamic estimation of worker reliabilities

So far we have assumed that the worker reliabilities, although probabilistic, are known to the assignment algorithm. Since the real-world is both uncertain and dynamic; hence, we relax the assumption of known reliabilities (Slivkins & Vaughan, 2013; Tarasov et al., 2014). To address this problem, the assignment algorithms can estimate a worker's reliability. A worker's reliability can be estimated through some test tasks before she joins the available worker pool; however, such approach cannot account for the dynamic changes in worker reliabilities and incurs extra travel costs (Tarasov et al., 2014). Alternatively, the worker reliabilities can be dynamically estimated over time through observed outcomes of assignments (Slivkins & Vaughan, 2013). In this regard, an online learning approach is required for estimating worker reliabilities as the assignment process progresses.

Fig. 3 highlights the online learning problem with help of example rounds. Worker reliabilities are estimated based on the outcomes of assignments in previous rounds. The learning problem becomes difficult due to the fact the algorithm can only observe the outcomes for previously chosen assignment set. The outcome of assignment is observed at the end of the round through a binary variable $y_{i,j} = Bernoulli(p_{i,j})$, where $p_{i,j}$ is the unknown worker reliability. The objective of an assignment algorithm is to adaptively optimize the MC-MRA problem while approximating worker reliability. In this adaptive assignment problem, the algorithm maintains and updates estimates of worker reliabilities $p_{i,j}$ over multiple rounds of assignment. We refer to this assignment problem, based on online learning, as the adaptive MC-MRA problem.

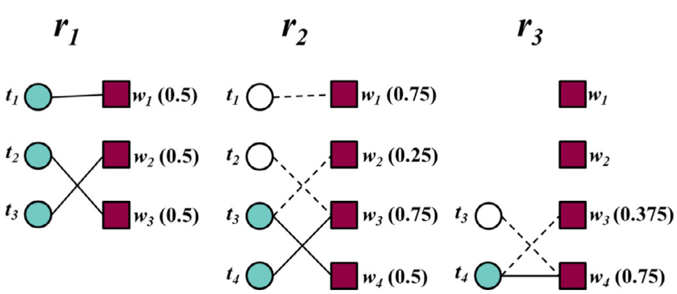

Fig. 3. Example of online learning for estimating worker reliabilities with semi-bandit feedback.

Definition 8. Let $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)$ be the vector of estimated reliabilities for all workers.

The assignment algorithm aims to minimize the difference between estimated worker reliabilities μ_j and actual worker reliabilities q_j. To this end, the algorithm must address the *exploration-exploitation* trade-off. On one side, the existing estimates of reliabilities can be used to make assignment decisions. This approach may be sub-optimal due to uncertainty of estimates. On the other side, the algorithm can deliberately choose sub-optimal workers to improve the accuracy of estimates.

The *multi-armed bandit* is a well-known problem that addresses the exploration-exploitation trade-off in sequential decision making (Barto, 1998; Powell, 2007). It formulates the trade-off faced by a player when repeatedly playing a k-armed slot machine. The player must maximize her cumulative rewards while pulling a arm in each round of play. The *combinatorial bandit* framework extends the multi-armed bandit problem to the choice of pulling a combination of arms in each round (Chen, Wang, & Yuan, 2013a). Combinatorial bandit problems are considered used three forms of feedback. *Full feedback* assumes that the rewards are observed for all arms irrespective of the chosen combination. *Semi-bandit feedback* means that the rewards are observed only for the chosen combination of arms . *Bandit feedback* means that only an aggregated reward value is observed based on the chosen combination of arms. We formulate the online learning problem, for estimation of worker reliabilities, according to the combinatorial bandit framework. Our proposed approach is referred to as the *semi-bandit learning* due to the fact the outcomes are observed only the chosen assignment set. Next we propose two algorithms based on different exploration strategies.

5.1. Greedy exploration approach

The greedy exploration approach is implemented in two phases; a random exploration phase followed by a pure exploitation phase (Auer, Cesa-Bianchi, & Fischer, 2002a). The basic idea is to randomly choose sub-optimal actions at the start to generate quick estimates of worker reliabilities. Afterwards, actions are chosen greedily based on the existing estimates while also learning from feedback. The duration of the exploration phase is controlled through an appropriate parameter.

Algorithm 2 details the complete assignment process for s number of rounds based on the DRR optimization and the greedy exploration. The algorithms maintains two variables for each worker k: the estimated reliability μ_k and the number of assignments to the worker θ_k. Each round starts with a listing of incomplete and active tasks along with available workers (Line 5–8). A distance matrix is calculated between incomplete tasks and available workers by calling the Distance sub-routine (Line 9). The algorithm requires a parameter ε that dictates how the reliability scores are calculated during each round. For the first ε percentage of rounds

Algorithm 2 The DRR–GRD algorithm

Require: $z, \varepsilon, \delta, \mathcal{T}, \mathcal{W}$
1: $M \leftarrow |\mathcal{W}|$
2: $[\mu_k]^M \leftarrow [0]^M$ {Initialize estimates}
3: $[\theta_k]^M \leftarrow [0]^M$ {Initialize counters}
4: **for** $r \leftarrow 1$ to s **do**
5: $T_r \leftarrow \text{Active}(\mathcal{T})$ {Set of incomplete tasks}
6: $W_r \leftarrow \text{Available}(\mathcal{W})$ {Set of available wokrers}
7: $n \leftarrow |T_r|$
8: $m \leftarrow |W_r|$
9: $[d_{i,j}]^{n\times m} \leftarrow \text{Distance}(T_r, W_r)$ {Distance matrix}
10: **if** $r \le \varepsilon \cdot z$ **then**
11: $[s_{i,j}]^{n\times m} \leftarrow [\mathcal{U}(0,1)]^{n\times m}$ {Random scores}
12: **else**
13: $[s_{i,j}]^{n\times m} \leftarrow \text{Greedy}(T_r, W_r, [\mu_k]^M)$ {Greedy scores}
14: **end if**
15: $[x_{i,j}]^{n\times m} \leftarrow \text{DRR}(\delta, [s_{i,j}]^{n\times m}, [d_{i,j}]^{n\times m})$
16: $X_r \leftarrow \{(t_i, w_j) \mid x_{i,j} > 0\}$ {Set of assignments}
17: $\text{Assign}(X_r)$ {Assign tasks}
18: $\text{Wait}(\tau)$ {Wait for end of round}
19: **for all** $(t_i, w_j) \in X_r$ **do**
20: $y_{i,j} \leftarrow \text{Complete}(t_i, w_j)$ {Completion indicator}
21: $\mu_j \leftarrow (y_{i,j} + \theta_j \cdot \mu_j)/(\theta_j + 1)$ {Update estimates}
22: $\theta_j \leftarrow \theta_j + 1$ {Update counters}
23: **end for**
24: **end for**

the score matrix is sampled using a standard Uniform distribution; therefore, resulting in random assignments for the purpose of pure exploration (Line 11). During the rest of the rounds, the score matrix is calculated based on the estimated reliabilities by calling the Greedy sub-routine (Line 13). The sub-routine approximates the worker reliability as $p_{i,j} \approx \mu_j$ and uses Eq. (4) to calculate the elements of the score matrix. The algorithm uses the DRR algorithm for solving MC-MRA problem using calculated distance and score matrices (Line 15). At the end of the round the algorithm observes the outcomes of the chosen assignments and updates the reliability estimates and counters variables accordingly (Line 19–23).

Note that the computational complexity of the DRR–GRD algorithm is bound by the complexity of the DRR algorithm. The performance of DRR–GRD algorithm in terms of approximating the actual worker reliabilities is dependent on the parameter ε. Understandably, very small values of ε may result in inaccurate estimates which may lead to sub-optimal exploitation. Conversely, large values may result in a high ratio of sub-optimal assignments due to over exploration.

5.2. Interval estimation approach

The interval estimation approach does not make any explicit distinction between exploration and exploitation. Instead actions are chosen optimistically by giving preference to options which have not been explored previously. The most widely known variant of this approach is based on the *upper confidence bound* (UCB) heuristic (Auer et al., 2002a). The basic idea is to calculate the confidence interval for the existing reliability estimates and define an upper bound for the expected values of estimates based on a confidence interval. During each round, actions are chosen by giving preference to higher upper confidence bounds instead of the actual estimates. The higher the uncertainty of the estimates the higher the chance of the worker being selected in a round. Subsequently, the uncertainty is reduced overtime due to this optimistic exploration.

Algorithm 3 The DRR−UCB algorithm

Require: $s, \delta, \mathcal{T}, \mathcal{W}$
1: $M \leftarrow |\mathcal{W}|$
2: $[\mu_k]^M \leftarrow [0]^M$ {Initialize estimates}
3: $[\theta_k]^M \leftarrow [0]^M$ {Initialize counters}
4: **for** $r \leftarrow 1$ to s **do**
5: $T_r \leftarrow \text{Active}(\mathcal{T})$ {Set of incomplete tasks}
6: $W_r \leftarrow \text{Available}(\mathcal{W})$ {Set of available wokrers}
7: $n \leftarrow |T_r|$
8: $m \leftarrow |W_r|$
9: $[d_{i,j}]^{n\times m} \leftarrow \text{Distance}(T_r, W_r)$ {Distance matrix}
10: $[s_{i,j}]^{n\times m} \leftarrow \text{UCB}(T_r, W_r, [\mu_k]^M, [\theta_k]^M)$ {UCB scores}
11: $[x_{i,j}]^{n\times m} \leftarrow \text{DRR}(\delta, [s_{i,j}]^{n\times m}, [d_{i,j}]^{n\times m})$
12: $X_r \leftarrow \{(t_i, w_j) \mid x_{i,j} > 0\}$ {Set of assignments}
13: $\text{Assign}(X_r)$ {Assign tasks}
14: $\text{Wait}(\tau)$ {Wait for end of round}
15: **for all** $(t_i, w_j) \in X_r$ **do**
16: $y_{i,j} \leftarrow \text{Complete}(t_i, w_j)$ {Completion indicator}
17: $\mu_j \leftarrow (y_{i,j} + \theta_j \cdot \mu_j)/(\theta_j + 1)$ {Update estimates}
18: $\theta_j \leftarrow \theta_j + 1$ {Update counters}
19: **end for**
20: **end for**

Algorithm 3 summarizes the assignment process based on the DRR optimization and the UCB exploration. Similar to Algorithm 2, it stores both estimates and counter variables for each worker. During each round, the score matrix is generated using the UCB subroutine (Line 10). For each assignment between task t_j and worker w_j, the reliability score is calculated by using Eq. (4) and approximating the worker reliability as follows:

$$p_{i,j} \approx \mu_j + \sqrt{\frac{3\ln(r)}{2\theta_j}}$$

where the second term quantifies the upper bound on the confidence interval for the estimated reliability of worker. The computational complexity of the DRR−UCB algorithm is also dominated by the DRR algorithm.

6. Empirical evaluation

We performed a set of experiments, on both real-world and synthetic data to evaluate the performance of the proposed approaches: MWBM, CDP, DRR, DRR-GRD, and DRR-UCB. First, we present our experimental methodology and then we present the results under various experimental settings.

6.1. Evaluation methodology

Data collection from large-scale deployments of prototypes is prohibitively expensive and time consuming, in SAT-based spatial crowdsourcing. Existing research works circumvent this issue by adopting datasets from location-based social networks (Deng et al., 2013; Kazemi and Shahabi, 2012; To et al., 2014, 2015; Yang, Zhang, Zheng, & Yu, 2015), mobile networks (Zhang et al., 2014), and urban transport systems (Cheng et al., 2015). We adopt a similar approach to evaluate the performance of our proposed algorithms. The evaluation of online algorithms with real users is also known to be notoriously difficult. In fact, evaluation using off-line simulations is common in existing literature on online algorithms (Li, Chu, Langford, & Wang, 2011). Following existing literature, we take a principled approach towards the evaluation of our algorithms using an *agent-based simulation* methodology (Schall, 2012; Zou, Gil, & Tharayil, 2014).

Agent-based simulation methodology helps to define the behavior of a spatial crowdsourcing environment based on individually defined agents in terms of their decision rules and communication. We defined three primary agents in the spatial crowdsourcing environment: requesters, workers, and platform. The simulation proceeds in rounds, where the number of rounds is fixed at the start. In the following we summarize the activities and interactions of each agent during a round:

- A *requester* agent dynamically submits new tasks to the platform during a round. In our simulation a requester is defined in terms of a task list. The list consists of a set of tasks distributed over a spatial region and a temporal time-line. Each task consists of three values: the start round, the spatial location, the expiry duration.
- A *worker* agent dynamically receives tasks from the platform at the start of a round and submits responses during a round. While performing tasks the worker also moves around in a spatial region that defines her spatial mobility. In our simulation a worker agent is defined in terms of a list of locations and reliability. The locations list is distributed over a spatial region and temporal time-line. Each location in the list also has an associated time when the worker visits the location. The average reliability of a worker agent is controlled by the parameter $q_j \in (0, 1)$.
- A *platform* agent provides the mediator's role between requesters and workers, while optimizing the reliability and travel costs. The assignment process is implemented in the platform agent that maintains a list of incomplete tasks and available workers. During each round, tasks are matched with workers using one of the assignment algorithm discussed earlier. The assignment algorithm queries the worker agents for the assigned tasks in a round and updates it knowledge according to the outcomes.

The goal of the evaluation is to demonstrate the effectiveness of the proposed algorithms under varying conditions. Next we outline the properties of the datasets used for the evaluation.

Datasets

We use a data-driven approach for initializing agents for simulation. For this purpose, we use both synthetic and real-world data to populate the variables of each simulated agent. The real-world dataset is based on data collected from a popular location-based social network: Foursquare.[1] The dataset contains check-ins, by people, at various locations in New York City from April 2012 to February 2013 (Yang et al., 2015). The dataset contains 1083 unique users, 38,333 unique locations, and 227,428 check-ins. A *check-in* represents the visitor relationship between a user and a location at a particular time. Fig. 4 shows the distribution of check-ins as a heat map, where red areas indicate higher concentration of check-ins. We assume that the users in the dataset are the crowd workers and initialize their mobility based on their individual check-ins. The locations are considered as the spatial tasks. We used the default settings from synthetic datasets for all other experiment settings, as described below.

Following existing methods for generating synthetic data (Cheng et al., 2015; To et al., 2015), the task and worker locations are uniformly distributed in a 2D space such that *latitute* $\sim$ *Uniform*(0, 1) and *longitude* $\sim$ *Uniform*(0, 1). The starting round for each task is also uniformly distributed such that $r_{st} \sim$ *Uniform*(1, 90). Similarly the reliabilities of workers are sampled from $q_j \sim$ *Uniform*(q_{min}, q_{max}). The expiry time for tasks is initialized to a fixed value such that $r_{et} = r_{st} + exp$. Table 1 lists the range of values for the algorithm parameters and other experimental settings, with default values in boldface font. We use Euclidean distance to quantify the travel costs between tasks and workers.

[1] https://foursquare.com/

Table 1
Experiment settings used for experimental evaluation using agent-based simulation. Default values for experiments are highlighted in bold.

Agent	Parameter	Description	Range of values
Worker	$[q_{min}, q_{max}]$	Range of the average reliability of a worker	[0.2,0.5], [0.2,0.6], [0.2,0.7], **[0.2,0.8]**
	m	Number of workers	50, **100**, 500, 1000
Requester	exp	Number of rounds between task arrival and expiry	1, **3**, 5, 7, 9
	r	Range of the assignment rounds	[1,50], **[1,90]**
	n	Number of tasks	500, **1000**, 2000, 5000
Platform	δ	Convergence parameter for DDR algorithm	0.01, **0.1**, 0.2, 0.3
	ε	Exploration parameter for DDR−GRD	0.05, 0.1, **0.2**

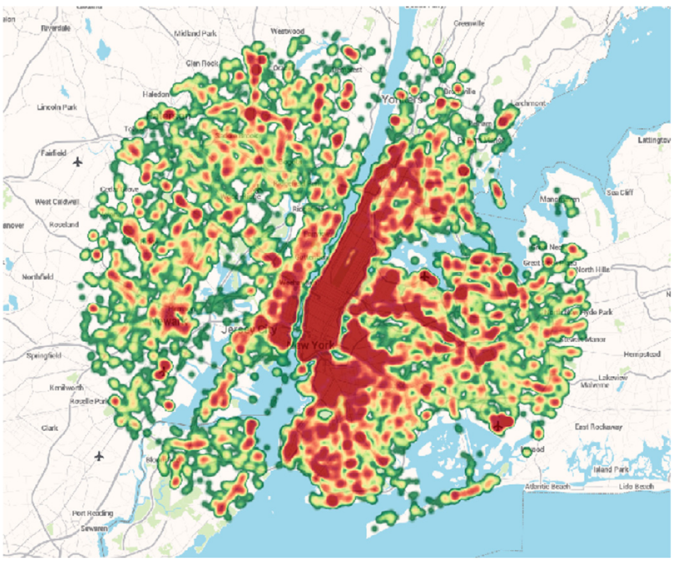

Fig. 4. Distributions of check-ins in the Foursquare dataset.

Evaluation metrics

We use four metrics for the evaluation of the assignment algorithms under different spatial crowdsourcing scenario (Cheng et al., 2015; Kazemi & Shahabi, 2012; To et al., 2014). The metrics are defined as follows:

- **Average reliability** is the mean of the reliability of all tasks at the end of all rounds. For each task, the reliability if the probability of completion the last assigned worker before expiry or completion of a task.
- **Average travel cost** is the mean of the distances to be traveled by workers assigned to tasks. For the purpose of reporting we only consider travel cost of the complete tasks. We measure travel cost for a task as the Euclidean distance between task and worker locations.
- **Assignments per task** is the number of assignment made for each task until it expires or completes. Lower reliabilities should result in less chance of task completion; therefore, resulting is higher number of assignments. Note that, the number of assignment is bound by the number of rounds between the start and expiry time for a task.
- **Task completion** is the percentage of tasks completed after the end of all rounds. From a requester's perspective, the task completion is the primary success criteria.

Experiment settings

The experiments where performed by varying a single parameter, while keeping others fixed. The experiment were run on an Intel Core i7-4600 CPU @2.90 GHz with 16 GB RAM. The algorithms were implemented using the open source libraries in Python. We used the Jonker and Volgenant variant of the Hungarian algorithm, as implemented in Python Materials Genomics (Pymatgen[2]) library for implementing MWBM, DRR, and CDP approaches (Ong et al., 2013). The linear programming phase of CDP approach was implemented using the Python Optimization Modeling Objects (Pyomo[3]) library (Hart, Laird, Watson, & Woodruff, 2012) and the GNU Linear Programming Kit (GLPK[4]). All reported metrics are based on the average of 10 runs for same dataset in an experiment.

6.2. Experiments using real-world data

The first set experiments compare the proposed algorithms against the baselines algorithms, on the real-world dataset. The reliabilities of workers are initialized based on the ratio of unique locations in a worker's check-ins against total number of locations. Fig. 5a shows the distribution of worker reliabilities. The majority of workers have low average reliability which is similar to the behavior of workers in a commercial platform Musthag and Ganesan (2013). The time duration of a round is initialized to a day. The simulation process proceeds by replaying the mobility of workers according to their respective check-ins in the dataset . In a round, the current location of a worker is set according to the her last check-in during the previous round. We uniformly sampled 50,000 check-ins to simulate tasks. The location associated with the check-in was considered the task location. The start time hence, the distribution of tasks locations and start time is the same as the distributions of check-ins in actual dataset. In summary, there are 322 rounds; 1083 workers; and 50,000 tasks for experiments with real-world dataset.

Fig. 5 b shows the comparison of algorithms in terms of the average reliability. Understandably, all the algorithms discussed in this paper perform better than the baseline RND algorithm. The MWBM, CDP, and DRR algorithms perform on similar levels, while the DRR-GRD and DRR-UCB algorithms achieve a 5–6% lower average reliability. The average reliability is low (between 0.27 to 0.33) for all algorithms. This could be explained due to the low reliability of worker population. In general, the performance of algorithms is somewhat similar when worker population is less reliable and less diverse.

Note that both CDP and DRR algorithms achieve high reliability with very small travel costs, as shown in Fig. 5c. In fact, the DRR algorithms and its learning based variants achieve even lower travel costs as compared to CDP, with relative decrease of almost 50%. Nonetheless, both CDP and DRR approaches can be used to minimize travel costs as a secondary optimization objective. In short, the our proposed DRR approach achieves much lower travel costs while maximizing reliability in comparison to the MWBM and the CDP approaches.

Fig. 5 d and Fig. 5e show the performance of algorithms in terms task completion after each round. The task completion is the ratio of the cumulative number of task completed against the

[2] http://www.pymatgen.org/
[3] http://www.pyomo.org/
[4] http://www.gnu.org/software/glpk/

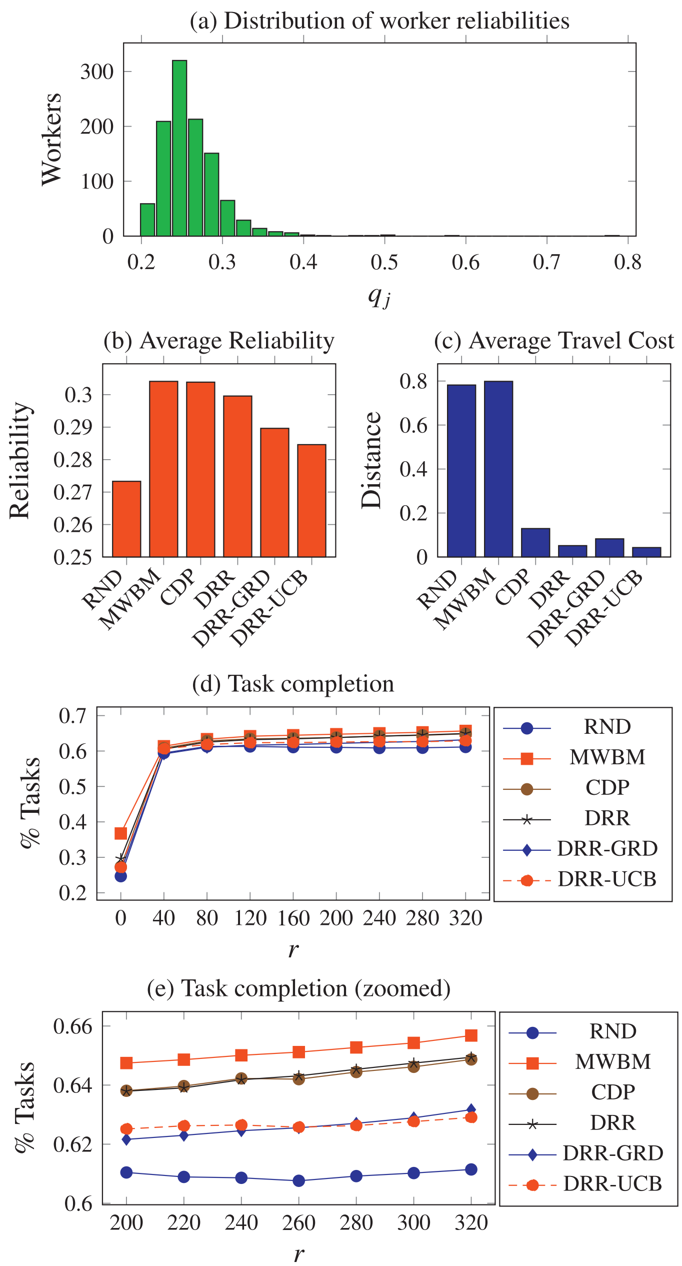

Fig. 5. Comparison of algorithms on Foursquare data for New York.

cumulative number of tasks appeared until round r. The MWBM achieves the best completions rate; however, the relative performance of other algorithms is slightly lower than MWBM. Note that the, task completion rate is very low in initial few rounds. This low completion rate is due to the skewed distribution of worker reliabilities. The task completion rate stabilizes after initial rounds, primarily due to repeated assignments of incomplete tasks. All algorithms achieve 70% less completion rate with expiry parameter *exp* set to three rounds for all tasks .

6.3. Experiments using synthetic data

The second set of experiments compare the algorithms using synthetically generate data. The goal of these experiments is to study the performance of algorithm under different conditions. In this regard, the simulation parameters are varied to study their effect on algorithm performance. In the following, we discuss the results of data generated with uniform distributions. The results for skewed distributions are provided in Appendix B.

Effects of expiry time

Fig. 6 shows the effect of varying the task expiration time *exp*, on the performance of the algorithms with known reliabilities. The increase in the expiry does not affect the average reliability of tasks across all algorithms, since it remains with a relatively stable range over all values (Fig. 6a). A similar pattern is observed for the travel costs of algorithms when compared to the increase in task expiry times (Fig. 6b). In terms of the relative performance of algorithms, the MWMB and CDP algorithms perform best in terms of reliability but they also perform worst in terms of travel costs. By comparison, the DRR algorithm although performs 10% less in term of reliability it achieves 80% less travel costs. This demonstrates the effectiveness of DRR algorithm against other algorithms with known reliabilities. The CDP algorithm fails in prioritizing small distances due to the uniqueness of solutions generated during the first phase of algorithm; therefore, the second optimization phase face limited choices with already high distances. The DRR algorithm performs optimization of the distance-reliability ratio; hence, achieving better results of both metrics.

Fig. 6 a and Fig. 6b also show the performance of the DRR-GRD algorithm. Note that, the DRR-GRD algorithm does not have access to the reliabilities at the time of assignment; instead, it approximates the reliabilities based on worker reliabilities estimated over time. We set the exploration parameter $\varepsilon = 0.2$ that controls the percentage of rounds with randomized assignment for the purpose of learning. The DRR-GRD algorithm achieves reliability within 10%–15% of the DRR algorithm while performing similarly on travel costs. This establishes the fact the DRR-GRD algorithm quickly estimates worker reliabilities that are exploited for assignments in later rounds. Fig. 6c and Fig. 6d show the number of assignments per task and the percentage of completed tasks for all algorithms. Understandably, the DRR and DRR-GRD algorithm require more assignments per task to ensure task completion. In the worst case, the DRR-GRD algorithm requires no more than 1.6 assignments even when expiry times are more that nine rounds for each task. Intuitively, the percentage completion of tasks reaches near maximum with expiry times of more than three rounds.

The variation in expiry times does not affect the performance in terms of reliability and costs; however, higher expiry times lead to better completion rates due to repeated assignments. In our simulations, the *exp* parameter is fixed at the platform level. In real deployment this parameter can be set by the requesters to individual tasks. Higher values of expiry times results in more tasks being offered to worker, primarily due to the repeated assignment of incomplete tasks.

Effects of workers' reliability

Fig. 7 shows the effects of the range of worker reliability on the comparative performance of the algorithms. We fixed the minimum value of the range $p_{min} = 0.2$ while varying the maximum value $q_{max} \in \{0.5, 0.6, 0.7, 0.8\}$ of the reliability of workers. The reliability increase linearly with the increase in the range of worker reliabilities, for all workers (Fig. 7a). Intuitively, the higher q_{max} results in more workers available with high success rate for tasks assigned to them. The distance-reliability ratio based algorithms consistently achieve lower travel costs with no effects due to the changes in worker reliabilities (Fig. 7b).

As expected, the number of assignment per task decreases with increase in q_{max}, indicating the availability of more reliable worker during each round. The relative performance, in terms of the number of assignments decrease constantly for all algorithms (Fig. 7c). Similar to the task reliability the percentage of completed tasks increases with an increase in the worker reliability range (Fig. 7d).

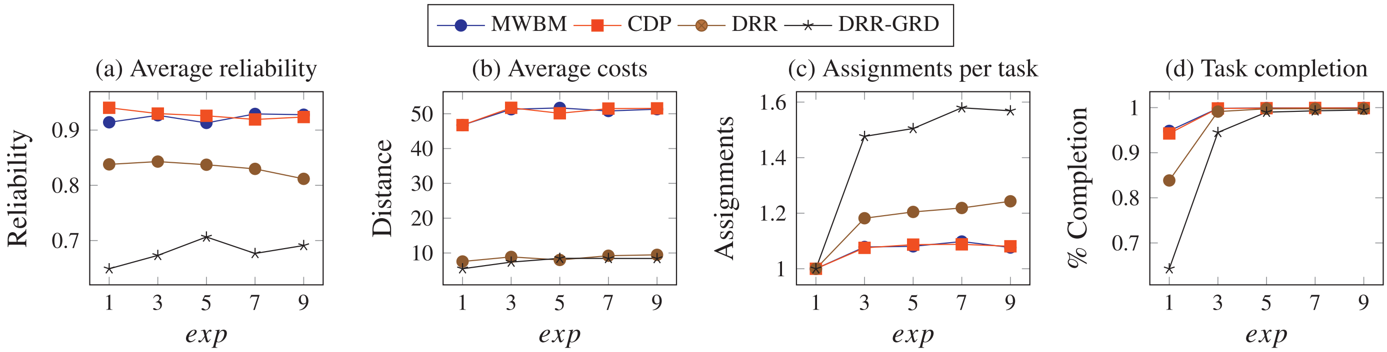

Fig. 6. Effects of task expiry time (Uniform data).

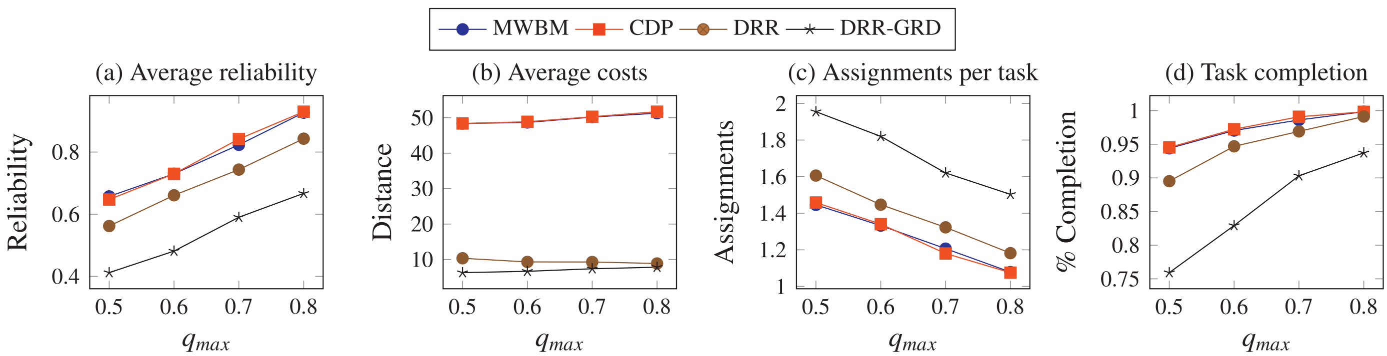

Fig. 7. Effects of the range of worker reliability, with $q_{min} = 0.2$ (Uniform data).

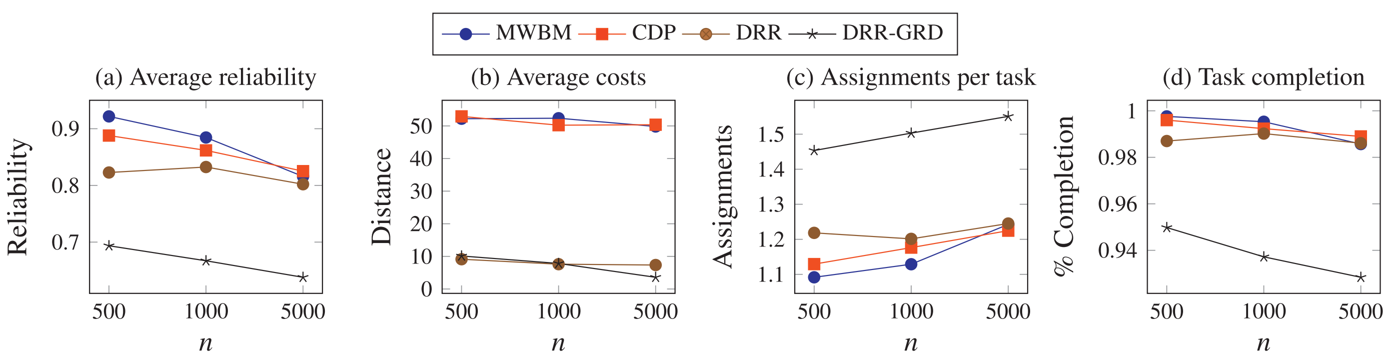

Fig. 8. Effects of the number of tasks (Uniform data).

Note that, even when the range of worker reliabilities is small [0.2, 0.5] the percentage of completed tasks is above 90% for algorithms with known reliabilities and above 75% for algorithm with estimated reliabilities. This indicates the effectiveness of the server-based assignment mode when workers are generally unreliable or reluctant.

Effects of number of tasks

Fig. 8 shows the effects of varying the number of tasks with parameter n; therefore, there are more tasks available for assignment during each round. The task reliability decreases slightly with an increasing number of tasks (Fig. 8a), while the average travel costs still remains the same even when there are ten times more tasks during a round (Fig. 8b). The decrease is primarily contributed due to low reliability workers being selected for more tasks in each round. Alternatively, if the worker reliabilities where skewed towards the higher end of the range [p_{min}, q_{max}] then the decrease might have been less. The performance of algorithms, in terms of the number of assignments per task, is the opposite of the reliability performance (Fig. 8c). The percentage completion of tasks also fall when the number of tasks per round increases. The rate of decrease is strongest for the DRR-GDR algorithm, possibly due to the randomized assignment during the exploration phase.

Effects of number of workers

Fig. 9 shows the effects of varying the number of workers with parameter m; such that, there are more alternatives available for assignment during each round. The average reliability of tasks increases due to more workers being assigned with higher q_j, as the number of workers increase (Fig. 9a). The relative rate of increase in task reliability is smaller for the DRR-GRD algorithm as compare to other algorithms. The MWBM and CDP algorithms achieve near maximum reliability for more than 500 workers. Note that, the CDP algorithm manages to reduce the average travel cost as the number of workers increases when compared to MWBM. This reduction is due to the close distance priority in the second stage of the algorithm while choosing an assignment from the feasible high reliability assignments. The number of workers does not have significant variation for the percentage completion of tasks (Fig. 9d).

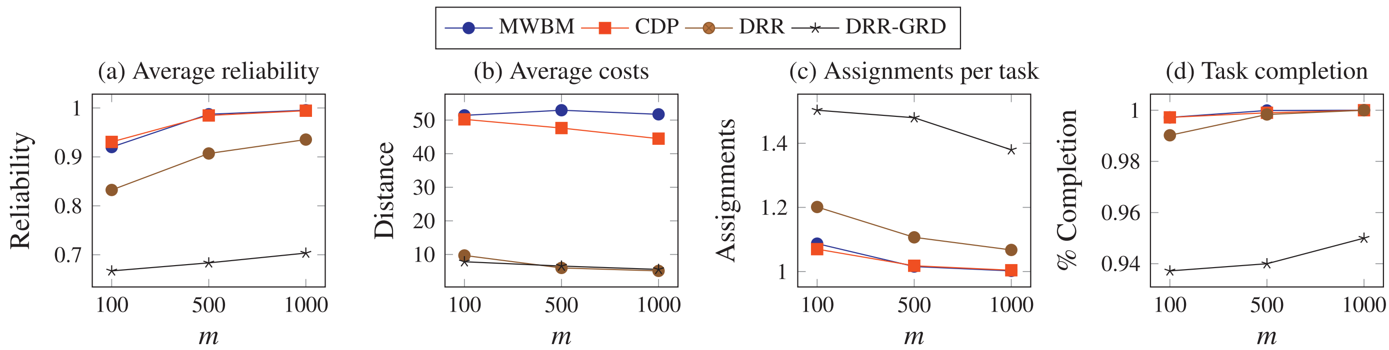

Fig. 9. Effects of the range of the number of workers (Uniform data).

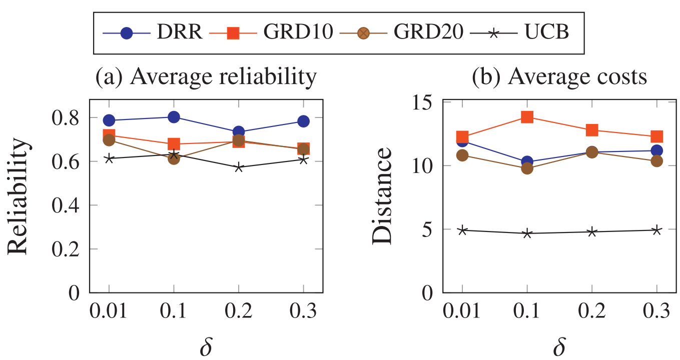

Fig. 10. Effects of the δ parameter of the DRR algorithm (Uniform data). UCB represents the DRR-UCB algorithm; whereas, GRD10 and GRD20 are variants of the DRR-GRD algorithm with $\varepsilon = 0.1$ and $\varepsilon = 0.2$, respectively.

Table 2
Comparison of algorithms in terms of execution time (Uniform data), where $r \in [0, 50]$, $n = 500$, $m = 100$, and other parameters with default values.

Algorithm	Execution time (seconds)
MWBM	0.66 ± 0.46
CDP	2.53 ± 0.59
DRR	**0.72 ± 0.45**
DRR-GRD	1.83 ± 1.01
DRR-UCB	0.95 ± 0.62

Table 3
Comparison of algorithms in terms of execution time (seconds) on Uniform dataset, where $r \in [0, 50]$, $n \in \{500, 1000, 2000\}$, $m \in \{50, 100, 200\}$, and other parameters with default values.

	$n = 500$		
Algorithm	$m = 50$	$m = 100$	$m = 200$
CDP	0.409 ± 0.10	1.524 ± 0.62	5.595 ± 2.59
DRR	**0.188 ± 0.12**	**0.564 ± 0.49**	**2.411 ± 2.78**
DRR-UCB	0.187 ± 0.10	0.707 ± 0.79	4.087 ± 4.59
	$n = 1000$		
	$m = 50$	$m = 100$	$m = 200$
CDP	0.726 ± 0.15	3.801 ± 1.01	14.338 ± 5.94
DRR	**0.526 ± 0.15**	**2.774 ± 1.21**	**11.925 ± 6.38**
DRR-UCB	0.625 ± 0.16	3.069 ± 1.33	16.384 ± 10.09
	$n = 2000$		
	$m = 50$	$m = 100$	$m = 200$
CDP	**2.184** ± **0.89**	9.318 ± 1.31	49.560 ± 14.87
DRR	2.481 ± 1.41	**8.356 ± 1.35**	**42.699 ± 10.50**
DRR-UCB	5.842 ± 3.95	10.108 ± 1.73	60.629 ± 17.35

Effects of algorithm parameters

Beside analyzing the comparative performances of algorithms, we analyzed the relationship between algorithm parameters and performance of algorithms. We varied the parameter λ for the algorithms based on Newton's method for distance-reliability ratio. Fig. 10 shows the comparison of the DRR, DRR-UCB and the DRR-GRD algorithm (using different values of the exploration parameter $\varepsilon \in \{0.10, 0.20\}$). The average reliability of tasks does not change significantly when the parameter λ takes on reasonably small values (Fig. 10a). The DRR-UCB algorithm achieves performance similar to the GRD-GRD algorithm in terms of the average reliability, specifically for $\lambda = 0.1$. The DRR-UCB achieves almost half of the travel costs achieved other algorithms. This results establishes the relative superiority of the DRR-UCB algorithm which does not require any parameter to control exploration. The UCB approach is also known to perform best asymptotically i.e. with infinite number of rounds (Audibert, Bubeck, & Lugosi, 2013).

Time performance comparisons

We also report the running times of our proposed approaches while fixing the number rounds $n = 50$, number of tasks $n = 500$, and number of rounds $r = [0, 50]$. Table 2 list the average execution time (in seconds) for a round, as well as the standard deviation. As expected, the MWBM algorithm performs best in terms of execution. The results also establish the relative performance gain of the DRR approach as compared to CDP approach. Although both approaches rely on the Hungarian algorithm as the base assignment approach, the parametric approach of DRR outperforms the two-phased approach of CDP. We used the open source GLPK solver for the second phase of CDP with linear programming, which results in higher execution times. Using commercial solvers such as CPLEX or GUROBI can improve the relative performance of CDP. The DRR-GRD algorithm takes more time when compared to the DRR algorithm due to the extra enumeration through feasible solutions when early estimates of worker reliability are similar. By comparison, the DRR-UCB algorithm performs similar to the DRR algorithm in terms of execution time. The relative higher computational time of the DRR-GRD algorithm, against the DRR-UCB algorithm, can attribute to the dense costs matrices generated by the initial random exploration.

Table 3 studies the scalability of the CDP, DRR, and DRR-UCB algorithms against different number of tasks and workers. As highlighted in bold, the DRR algorithm performs the best in general. Interestingly the DDR-UCB algorithm performs the worst when number of tasks and workers is high. This could be due to the extra time taken by the DRR subroutine to converge.

7. Discussion

In this section, we discuss the implications of the comparative performance results of the algorithms discussed in the paper. We categorize the discussion in terms of the two primary aspect of the MC-MRA problem: optimization and learning.

7.1. Formulation of optimization objectives

This paper presents three approaches to addressing the MC-MRA problem. The CDP and DRR approaches optimize the reliability as well as the travel costs. Our proposed DRR approach is based on the linear-fractional programming formulation of the MC-MRA problem. The linear-fractional programming approach for optimization is a generalized case of the linear assignment problem; therefore, our approach provides a more generalized solution to the MC-MRA problem. Other cost-reliability ratio assignment problems, in crowdsourcing, can also be formulated following a similar approach. Since the approach transforms the fractional optimization objective to an equivalent linear optimization objective, any existing algorithm for linearized solution can be used to solve the problem. Subsequently the computational complexity is dominated by the problem size and the computational complexity of the algorithm used for linearized solution.

The MC-MRA problem presented in this paper assumes one response constraint per task, which means that the each task is assigned to at most one worker. Consideration of multiple workers per tasks either for redundancy or diversity is a possible extensions; however, such constraints increase the complexity of problem requiring approximate solutions. We limit the scope of this paper to keep the assignment problem tractable, while keeping further extensions as future work. As a simple extension, each task that requires multiple workers can be considered as the multiple instances of the same task. Similarly, a worker with more than one task capacity can be modeled as multiple instance of the same worker. The performance of the approaches proposed here need to be validated with more complex constraints, such as the task diversity (Cheng et al., 2015) or budget constraints (Tran-Thanh et al., 2014). Even worker constraints such as capacity or time duration can further enhance the assignment process. Note that, a linear-fractional program can be easily transformed to an equivalent linear program as far as the constraints matrix is unimodular on the right hand side (Bajalinov, 2003).

7.2. Online learning heuristics

Besides the DRR approach, the other main contribution of this paper is the estimation of worker reliability in the case of uncertainty. The real world is uncertain; therefore, it is essential to incorporate appropriate learning capabilities in spatial crowdsourcing process. The real world is dynamic; hence, it is useful to update the learning in spatial crowdsourcing process. Our approach for the dynamic estimation of workers' reliabilities is based on the multi-armed bandit model, which deals with uncertain decision making over time. Among the two learning approaches proposed here for the adaptive MC-MRA problem, the greedy approach performs reasonably well in relation with the deterministic algorithms. However, the greedy approach suffers from short term performance as the learning is scheduled in first few rounds. To overcome this issue, an alternative greedy approach based on semi-uniform learning is proposed in literature where randomized exploration is scheduled during any round based on a probability parameter (Auer et al., 2002a). Both of these greedy approaches suffer from randomization error. By comparison, the upper confidence bound approach performs better in the long-term which underline its utility for long running systems with dynamic worker populations.

Both DRR-GRD and DRR-UCB algorithms estimate the reliabilities of workers to approximate the worker reliabilities in each round of assignment. The exploration is based on the estimates plus an adjustment term that depends on the number of times a worker has been chosen previously. This means that the more a worker is explored the more accurate the estimates are. Instead of estimating distribution parameters, we can also estimate the actual distribution of the characteristic by using Bayesian methods (Scott, 2010). Similarly, probability matching approaches try to match the number of times a worker is chosen with the probability of that worker being the most reliable (Scott, 2010; Vermorel & Mohri, 2005).

Beside the feedback of task completion, other contextual information helps to improve the learning process by providing additional data points. Existing approaches based on the multi-armed bandit problem tend to exploit this information for the purpose of improving the estimates of future rewards (Li, Chu, Langford, & Schapire, 2010). In our current formulation the locations of tasks and workers serves as the contextual information; however, we exploit this information to optimize travel costs. Contextual algorithms (Hassan & Curry, 2014) exploit side information including locations and other information such as task types, worker demographics, location properties, etc. Utility of such approaches in combinatorial optimization setting is yet to be investigated for spatial crowdsourcing.

7.3. Reliability in spatial crowdsourcing

The worker reliability as defined in this paper depends on both task and worker. This definition of reliability does not include the effects of external factors on the probability of success. It also does not consider the task or requester specific criteria for successful task completion. Yet, the same problem formulation is applicable in scenarios when the success criteria is binary. For instance, a requester might define the successful completion of a task in terms of the resolution of the pictures uploaded by the worker. On the other hand, the other requester might define the success in terms of a presence of specific objects in uploaded pictures. In this sense, the reliability of a worker might be thought of as the rate of successful task completion. Therefore, it is intuitive to maximize the reliability through intelligent assignment decisions.

The probabilistic definition of reliability is fairly recent in spatial crowdsourcing literature. The reliability is defined as the probability of task being completed by an assign worker. A recent proposal defined probability as decreasing function of the distance between task and worker (To et al., 2014). Additionally, it was assumed to be independent of task and worker. Another proposal defined fixed probability as the reliability of a worker (Cheng et al., 2015) . The reliability in terms of the frequent routes visited by a worker was also considered for WST-based crowdsourcing Chen et al.. In general, all these definitions of reliability are independent of tasks. Instead, this paper assumes worker reliability being dependent on the assigned task.

Within the literature on non-spatial crowdsourcing, the reliability of a worker is defined in terms the accuracy of responses provided by the worker. On one hand, the reliability of worker can be the closeness of response provided by a worker to the predicted values of a regression task (Tarasov et al., 2014). On the other hand, it is the ratio of correct responses against all responses provided by a worker for binary classification tasks (Ho et al., 2013). Domain specific definitions of worker reliability are common in literature on non-spatial crowdsourcing. However, the definition of worker reliability in this paper is domain agnostic.

7.4. Limitations and strengths

One of major strength of the proposed approach is its applicability to more realistic spatial crowdsourcing scenario, as compared to existing task assignment approaches in spatial crowdsourcing. The platform being an participating agent in the multi-agent environment of spatial crowdsourcing, must adapt its behavior according the outcomes of task assignments overtime. In this regard, our proposed approach enable intelligent assignment decisions for optimizing spatial crowdsourcing, in the face of uncertainty. Our proposed approach is also applicable to other areas of expert and intelligent systems. It is specifically suited for application scenarios when optimization objective is to maximize cumulative rewards and minimize cumulative costs, when rewards are uncertain.

The adaptive task assignment approach faces limitations due to three underlying assumption. First the multi-armed bandit formulation of the reliability approximation necessitates immediate observability of assignment outcomes. Second, it is assumed that the worker reliabilities are stochastic. This assumption ignores strategic behavior of worker in responses of previously assigned tasks. Learning heuristics for adversarial outcomes of decisions have also been studied in literature on multi-armed bandits. However, a detail investigation of such heuristics, for task assignment in spatial crowdsourcing, is out of scope of this work. Finally, there could be situations when worker might attempt an assigned task after end of a round. Incorporating such delayed outcomes is another limitation of this research work.

Assigning a chain of tasks to a worker such that the assigned tasks are clustered within an area is also intuitive Musthag and Ganesan (2013). The DRR algorithms assign nearest tasks to a worker over many rounds, which can also form a chain of tasks over time. However, a worker cannot plan travel path since future tasks are not revealed beforehand. Task chains can be considered in case of multiple tasks per worker in a round. But this is a known hard problem and out of scope of this work. Furthermore, it adds to the complexity of assignment decisions due to considerations of load balancing and social welfare. Social welfare entails that all workers are given opportunities to perform tasks, instead of favoring workers who adversely specify their load capacity to get more tasks. The goal of optimizing social welfare is to promote long term engagement Teodoro, Ozturk, Naaman, Mason, and Lindqvist (2014).

8. Conclusion and future work

This paper extends the existing research on spatial crowdsourcing in several key ways. Firstly, this work provides a conceptual framework to study the *minimum-cost maximum reliability assignment problem* with online combinatorial optimization and online learning. It highlights the key aspects of the assignment algorithms in stochastic and online settings. This framework can be used to align further research contributions in crowdsourcing. Secondly, this work provides new insights into the combinatorial assignment strategies when the objective is to maximize reliability and minimize costs. It provides evidence of effectiveness of the adaptive assignment algorithms with uncertain reliabilities and deterministic costs. A *distance-reliability ratio* based assignment approach is proposed to maximize the reliability of spatial tasks while minimizing the travel costs. Experimental evaluation provides evidence of the effectiveness of the proposed approach against existing approaches. The proposed approach achieves more than 80% decrease in travel costs in comparison to previously known approaches, while achieving similar reliability for spatial tasks.

We extend the *distance-reliability ratio* approach with two online learning algorithms for estimating unknown worker reliabilities. The first *greedy exploration* approach achieves travel costs similar to the *distance-reliability ratio* approach with known reliabilities, while achieving reliability within 15%. The second *interval estimation* approach achieves even better travel costs when compared to the *greedy exploration* approach. In general, the results suggest the online learning based assignment algorithms perform reasonably well even under varying conditions. As part of the future work, we plan to extend our work with complex constraints. For instance, instead of a fixed cardinality constraint on the assignment of worker to a task, the cardinality can be a dynamic function depending of the application domain. We also aim to extend this work to assignment problems with multiple tasks to workers while considering their mobility trajectories and preferences in terms of spatial divergence from their trajectories.

Acknowledgments

This work has been supported in part by the Seventh EU Framework Programme (FP7) from ICT grant agreement No. 619660 (WATERNOMICS) and the Science Foundation Ireland (SFI) under grant No. SFI/12/RC/2289.

Appendix A. Background

Crowdsourcing has emerged as a powerful paradigm for solving complex problems at large scale with the help of a group of people (Hassan & Curry, 2013; 2014; Kittur et al., 2013). The rapid development in web technologies have made it possible for millions of online users to contribute towards specific problems. People can contribute by performing tasks such as collecting photos, transcribing audio, classifying images, classifying items, etc (Kittur et al., 2013). A crowdsourcing system, in general, has three types of interacting agents: requesters, workers, and platform. Each of these agents is described as follows:

- *Requesters* submit tasks, to the platform, that need to be performed by the crowd. Apart from humans, the requester can also be another application that needs human services for performing it functionality. Requesters are generally interested in maximizing their utility that is generally defined in terms of the quality of task performance and the associated costs. Note that, the notion of quality and costs can vary between types of tasks and the application domain.
- *Workers* are the members of the crowd who are willing to perform tasks. Workers can vary in terms of their reliability of performed tasks and the incentive they expect against the work. Worker are generally interested in maximizing their own utility that is defined in terms of the effort they exert and value they gain from performing tasks.
- *Platform* is a software that serves as the mediator between requesters and workers; therefore, providing the interaction mechanism between both agents. It defines the mode of exchange for tasks, results, feedback, and incentives. A third-party platform provider is generally interested in maximizing the value gained from the use of the software and its functionality. Furthermore, it is in the interest of platform managers to promote long-terms use of their platform.

Fig. A1 highlights the sequence of interactions between these agents. The requester submits tasks to the platform which allows filtering of workers based on their characteristics or categories. The tasks are assigned to the appropriate workers. The workers perform the tasks and submit the responses to the platform. The platform assembles the results of crowdsourcing by aggregating and filtering the responses depending on the application domain. The results are sent back to the requesters. Spatial crowdsourcing includes tasks that are situated in physical world and require workers to travel to their associated locations. An *offline* assignment

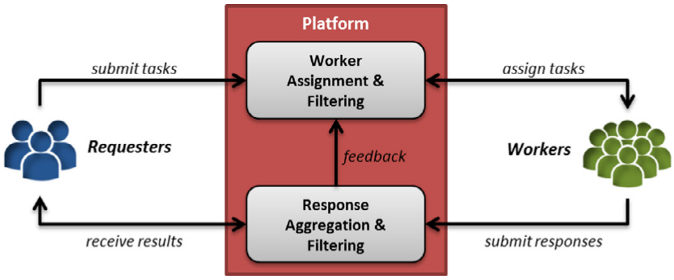

Fig. A1. An overview of typical interaction between agents in crowdsourcing.

involves matching the all tasks with all available workers, in single batch process. Such a batch process is unrealistic in a multi-agent system like crowdsourcing due to dynamic nature of tasks and workers. By comparison, an *online* assignment entails dynamically matching tasks with worker over time.

Dynamic assignment

It is natural for the agents to interact repeatedly over time in spatial crowdsourcing. During such interactions each agent is involved in sequential decision making with the aim of optimizing its own utility (Slivkins & Vaughan, 2013). The repeated interaction means that the agents can learn about each other and adjust their behavior, accordingly. For instance, a worker can decide whether to perform a task given its value and required effort. The platform can filter workers or requesters to discourage malicious behavior and promote long-term use. A requester can choose a worker for each task such that the her utility is maximized. The utility of crowd work is dependent of who performs the task, how good the task is performed, and how the results are combined. Since workers can be heterogeneous in terms of their reliability and costs, an appropriate assignment process has direct impact of the utility of crowd work (Kittur et al., 2013; Slivkins & Vaughan, 2013). Differences of reliability means the workers might have different rate of task completion with high quality within time. In case of spatial crowdsourcing, the heterogeneity is further exasperated due to the spatio-temporal context of tasks and workers. This paper primarily focuses on intelligent approaches of dynamic assignment for spatial crowdsourcing. Dynamic assignment is a sequential decision making problem that aims to iteratively match tasks with workers, under uncertain conditions (Kittur et al., 2013; Law & Ahn, 2011; Slivkins & Vaughan, 2013). Rest of this section is dedicated to summarizing the existing research on dynamic assignment problems in spatial crowdsourcing.

Dynamic assignment, in spatial crowdsourcing, is generally realized under either a *real-time* or a *periodic* assignment protocol. The real-time assignment protocol means that the tasks are matched with workers either on task arrival or worker arrival. In case of the assignment protocol for task arrivals, the decision involves choosing a set workers from a the pool of available workers (Hassan & Curry, 2014; To et al., 2014; Tran-Thanh et al., 2014). The other real-time assignment protocol assumes that worker dynamically arrive on the platform and the platform then chooses a task (or set of tasks) for each worker from a pool of available tasks (Ho et al., 2013; Ho & Vaughan, 2012; Karger, Oh, & Shah, 2011b). The periodic assignment protocol assumes that to match sets of tasks and available after regular intervals of time (Cheng et al., 2015; Kazemi and Shahabi, 2012; To et al., 2015).

Methods of dynamic assignment

Kazemi et al. proposed a taxonomy of spatial crowdsourcing that highlights two modes of task assignment: *worker selected tasks* (WST) and *server assigned tasks* (SAT). The majority of general crowdsourcing platforms employ WST for task assignment (Chen et al., 2013b; Deng et al., 2013; Difallah, Demartini, and Cudré-Mauroux, 2013; Ho et al., 2013; Ho & Vaughan, 2012; Karger et al., 2011b; Law & Ahn, 2011). In this method, workers visit the crowdsourcing platform and self-assign tasks through an appropriate search and browse interface. The WST is characterized by high emphasis on self-determination of tasks to perform, by the workers. The WST method is prone to search friction, possibly due to the explicit interaction required from workers (Kulkarni et al., 2012). Search friction arises when workers have difficulty finding the right tasks, or vice versa. Task recommendation techniques such as collaborative filtering and context-based filtering have been proposed to address these issue of WST (Geiger & Schader, 2014).

The SAT method addresses this issue by algorithmically managing the selection process (Hassan & Curry, 2013; 2014; Ho & Vaughan, 2012; Kazemi & Shahabi, 2012; Slivkins & Vaughan, 2013; To et al., 2014). The SAT method relies upon the knowledge about tasks and workers to find out suitable matches. Besides the task assignment methods, the taxonomy describes the roles of monetary incentives and task redundancy for improving the utility of crowd work. This paper has primarily focused on SAT-based task assignment for self-incentivized spatial crowdsourcing with single worker per task.

Adaptivity in dynamic assignment

Dynamic assignment, under deterministic settings, assumes that the platform has full knowledge of task and worker characteristics at the time of assignment (Karger, Oh, & Shah, 2011a; Law & Ahn, 2011). In other words, the expected outcomes of assignment decisions are known beforehand for each alternative. Effectively the assignment process involves finding a matching between tasks and workers that is locally optimal in each round. The real-world non-deterministic, as workers might not complete tasks assigned to them. Therefore, the assignment decision must be made under non-deterministic settings with partially observable feedback about assignment outcomes.

Uncertainty arises due to the fact that the probability of task completion by a worker is a stochastic process. The optimization objective of dynamic assignment is to match tasks with workers such that the probability of task completion is maximized. Partial observability means that the assignment outcomes are observed only for the selected task-worker pairs. In situations where the probabilities of task completion are unknown, the must adaptively must estimate the probabilities by learning from the outcomes of previous assignments. Such *online learning* problems have been studied under two settings in literature (Barto, 1998; Powell, 2007): *full feedback* and *bandit feedback*. Full feedback means that the reliability variables are observed for all workers after assignment (Bubeck & Cesa-Bianchi, 2012). Bandit feedback means that the reliability variables are only observed for chosen workers after assignment (Bubeck & Cesa-Bianchi, 2012). The dynamic assignment while learning is known as the *adaptive assignment* problem in the literature (Slivkins & Vaughan, 2013). A fundamental challenge of adaptive assignment with partial feedback is to balance the *exploration-exploitation* trade-off (Barto, 1998; Shalev-Shwartz, 2011).

An assignment algorithm could repeatedly choose task-worker pairs which seem to optimize the defined objectives, also known as *exploitation* (Barto, 1998). Due to the uncertainty of existing knowledge, the seemingly best set of pairs might be the suboptimal choice. Alternatively, the algorithm might choose pairs for the purpose of *exploration* (Barto, 1998). In such a case, the algorithm deliberately makes a suboptimal choice for the sake of learning. The exploration might adversely affect the optimization objective in the

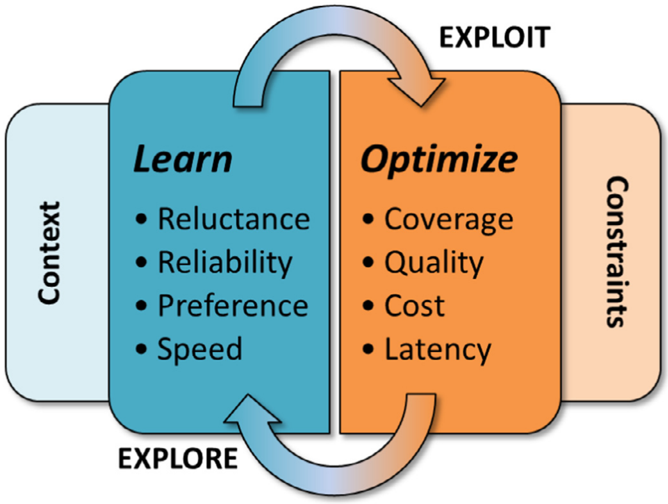

Fig. A2. A conceptual overview of primary aspects of the dynamic assignment in crowdsourcing environments.

short term, but the additional knowledge can help improve assignment choices in the long-term. Understandably, neither a pure exploration nor a pure exploitation strategy can produce the best results. A good assignment algorithm strikes the right balance between exploration and exploitation (Kittur et al., 2013; Law & Ahn, 2011). The exploration-exploitation trade-off have been formalized and studied under multi-armed bandit problem in existing literature (Auer et al., 2002a; Auer, Cesa-Bianchi, Freund, & Schapire, 2002b; Berry & Fristedt, 1985; Gittins, Glazebrook, & Weber, 2011; Gittins, 1979).

Dimensions of dynamic assignment

The specifics of the assignment process are driven by the design considerations including task design, platform design, and performance objectives. The design considerations are in turn dependent on the requirements of the application scenario. Nonetheless, there are some fundamental dimensions of *dynamic assignment* under non-deterministic settings with partial feedback. Fig. A2 highlights these dimensions which are further described below:

- *Optimize:* The first primary dimension of dynamic assignment is the optimization of an objective function. The assignment process can be designed to optimize four different types of objective: coverage, quality, costs, and latency. Coverage defines the numbers of tasks assigned (Kazemi & Shahabi, 2012) or completed (To et al., 2014). Quality defines the number of tasks that meet the utility criteria of the requester. For instance, the aim of crowd sourced machine learning is to collect labels that have the highest improvement in the prediction accuracy of classification model (Dekel, Gentile, and Sridharan, 2012; Ho et al., 2013).
- *Constraints:* The second dimension covers the various constraints enforced by the requesters and/or workers. In general, the requesters can put budgetary and/or time constraints (Tran-Thanh et al., 2014). Workers can impose cost and/or capacity constraints (Ho et al., 2013). Task specific constraints can also exist in crowdsourcing; for instance, the spatial constraints on area covered by workers (Kazemi & Shahabi, 2012).
- *Learn:* The third dimension is concerned with the learning process for estimating worker characteristics. Crowd workers can vary in terms of their characteristics such as reluctance, reliability, speed, and preferences. Gathering and exploiting knowledge on these characteristics can help improve the assignment process. For instance, the reluctance levels of workers can vary depending on the travel distance for spatial tasks (Teodoro et al., 2014; To et al., 2014). Similarly, some workers can achieve a higher speed of work as compared to others.
- *Context:* The fourth dimension concerns the contextual information of tasks and workers. For instance, the context of a spatial task is completely different from the classification task. The nature and availability of contextual information can help improve the assignment process. The contextual information can include taxonomy of tasks (Difallah et al., 2013; Hassan & Curry, 2013; Hassan, O'Riain, & Curry, 2013), categories of workers (Abraham et al., 2013), and spatial attributes of tasks and workers (Hassan & Curry, 2014). The usefulness of the contextual information varies depending on the nature of crowdsourcing.

Considering the methods, adaptivity, and dimensions, a set of research requirements were identified for dynamic assignment in ubiquitous crowdsourcing. These requirements help in identification of the expected contributions of this paper.

Existing literature

We classify the existing literature on dynamic assignment, in crowdsourcing, into two broad categories of tasks: Non-spatial and spatial. In the following, we summarize each these and their specificities:

Non-spatial tasks. Table A1 compares existing literature, in terms of the primary dimensions, on dynamic assignment in non-spatial crowdsourcing. Majority of the research work for non-spatial crowdsourcing is focused on improve the quality of responses submitted by workers, using dynamic assignment. The quality of responses is defined as the correctness of responses submitted for classification tasks (Ho & Vaughan, 2012), regression tasks (Tarasov et al., 2014), or survey tasks (Abraham et al., 2013). Therefore, some works have proposed adaptive assignment approaches based on the estimated reliabilities of workers (Abraham et al., 2013; Ho et al., 2013; Tarasov et al., 2014). The primary difference between these works has been in terms of the problem formulation and the types of constraints.

Ho and Vaughan formulated the adaptive assignment problem for single requester with constraints on the number of tasks for each task type (Ho & Vaughan, 2012). They consider the dynamic assignment with workers arrivals following a stochastic process and proposed an assignment algorithm based on estimated worker reliabilities. Ho et al. extended their previous work with worker capacity constraints (Ho et al., 2013). Further they considered the situation where assignment decision are made in conjunction with estimation of correct responses. By comparison, an recent approach considered the situatio Tarasov et al. proposed an dynamic assignment framework, based on multi-armed bandit approach, for estimation of worker reliabilities. Their approach is differentiated from above due to the fact that it does not require any gold standard tasks for learning. All these works assume predefined set of tasks with dynamic arrivals of workers; hence, the assignment decision involves choosing a set of tasks for the current worker to perform.

There have been recent research proposals that formulate dynamic arrivals of tasks that are assigned to a pool of workers. Specifically, the *bandit survey problem* considers assignment of survey tasks to different crowds Abraham et al. (2013). The proposed formulation address both task assignment as well as optimal stopping problems for survey tasks. Tran-Thanh et al. proposed the *bounded multi-armed bandit* problem that considers budget limits for the multi-armed bandit problem with multiple plays Tran-Thanh et al. (2014). The proposed problem was mapped to the expert crowdsourcing scenarios where each tasks involves complex development activities.

Table A1
Survey of the existing literature on dynamic assignment with server-assignment tasks in crowdsourcing.

Source	Optimize	Constraints	Learn	Context	Dynamic arrivals
Ho and Vaughan (2012)	Response quality	Task redundancy	Worker reliability	Task types	Workers
Ho et al.(2013)	Response quality	Worker capacity	Worker reliability	Task types	Workers
Tarasov et al. (2014)	Response quality		Worker reliability	Regression tasks	Workers
Dekel et al. (2012)	Response quality	Task redundancy	Worker reliability	Classification tasks	Workers
Abraham et al. (2013)	Response quality		Worker reliability	Survey tasks	Tasks
Tran-Thanh et al. (2014)	Tasks coverage	Incentives budget	Worker reliability	Complex tasks	Tasks

Table A2
Survey of the existing literature on dynamic assignment with server-assignment tasks in spatial crowdsourcing.

Source	Optimize	Constraints	Learn	Context	Dynamic arrivals
Kazemi and Shahabi (2012)	No. of assignments Travel distance	Worker capacity Spatial region		Spatial tasks	Tasks Workers
To et al. (2014)	Tasks reliability			Spatial tasks	Tasks
Deng et al. (2013)	Tasks coverage			Spatial tasks	Workers
Hassan and Curry (2014)	Tasks coverage	Task redundancy	Worker reluctance	Spatial tasks Review tasks	Tasks
To et al., (2015)	Expertise score Travel distance	Worker capacity Task redundancy Spatial region		Spatial tasks	Tasks Workers
Cheng et al. (2015)	Tasks reliability Tasks diversity	Worker velocity		Spatial tasks	Tasks Workers
Our approach	Tasks reliability Travel distance	Worker capacity Task redundancy	Worker reliability	Spatial tasks	Tasks Workers

Apart from being focused on the spatial crowdsourcing scenario, our proposed MC-MRA problem is differentiated from these works from three key perspectives. First our formulation considers dynamic arrivals of both tasks and workers; whereas, most of the existing approaches for non-spatial tasks are limited to dynamic arrival for either tasks or workers. Second we aim to optimization both reliability and costs. By comparison, above discussed proposals only optimize either reliability or costs. Third, we consider the situation where the number of rounds in unknown in advance.

Spatial tasks. Spatial crowdsourcing entails relatively different assignment approach due to the spatio-temporal nature of tasks and longer durations of time required to performing tasks. Table A2 compares existing literature on the dynamic assignment in spatial crowdsourcing. Kazemi and Shahabi proposed the *maximum task assignment* problem for spatial crowdsourcing, which was later extended to *maximum score assignment* problem (Kazemi and Shahabi, 2012; To et al., 2015). Both proposals are based on deterministic settings for the assignment while maximizing the number of assignment and expertise scores, respectively. To et al. defined at privacy enabling framework for task assignment in spatial crowdsourcing (To et al., 2014). The framework is designed to hide the actual locations of workers during assignment process. Deng et al. (Deng et al., 2013) propose approximation algorithms for scheduling task for worker selected tasks. The proposed algorithms aim to maximize the number of task performed by an individual worker. All of these proposals do not consider the real-world situation of uncertain outcomes of assignments; whereas, we consider stochastic assignment settings with semi-bandit feedback.

Recent proposals have considered the dynamic assignment problem in probabilistic for spatial crowdsourcing (Chen et al., 2014; Cheng et al., 2015; Hassan & Curry, 2014). One proposal focused on maximizing diversity of data collected from spatial tasks (Cheng et al., 2015). Other proposal consider the online learning problem for estimating worker reluctance for spatial crowdsourcing (Hassan & Curry, 2014). Another proposal consider task recommendation based on the previous trajectories of dynamically arriving worker (Chen et al., 2014). By comparison, our work is differentiated from these proposal along two key aspects. These proposals are limited to optimization of reliability; whereas, we formulate the joint optimization reliability and costs. Apart from (Hassan & Curry, 2014), none of the proposal consider the online learning aspect of adaptive assignment problem. Hassan and Curry focus on contextual learning models with single assignment per round. In comparison to (Hassan & Curry, 2014), we consider more generalized setting for combinatorial assignment in each round.

Another class of crowdsourcing focuses on exploiting people with mobile phones for collection sensing data at various locations (Zhang et al., 2014). Optimal task allocation problem under time constraints has been addressed recently (Feng et al., 2014). Xiao et al. propose a multi-task assignment approach for workers from mobile social networks (Xiao et al., 2015). Other approaches include assigning tasks based on worker trajectory, also known as the orienteering problem (Chen et al., 2014; Chen et al., 2015). None these worker address the adaptive assignment problem; therefore, the techniques discussed in the paper are complimentary to these works.

Appendix B. Results of skewed distributions

In this section, we discuss the experimental evaluation of proposed algorithms on skewed distributions. Give the range of worker reliabilities $[q_{min}, q_{max}]$, we sampled worker reliabilities from Normal distribution i.e. $q_j \sim Normal((q_{max} - q_{min})/4 + q_{min}, (q_{max} - q_{min})/4)$. In this following, we provide performance results of algorithms under various experimental settings.

Effects of expiry time

Fig. A3 shows the effect of varying the task expiration time δ, which are similar to the results for the uniform distribution of worker reliabilities. In general the average reliability is lowered due to the smaller values of worker reliabilities (Fig. A3a). The DRR-GRD algorithm improves in terms of average reliability per task as the expiry times increases. Algorithms based on DRR approach perform significantly better in terms of the travel costs (Fig. A3b). Fig. A3c shows the number of assignment per task for each algorithm. Understandably, the overall performance of all algorithms is

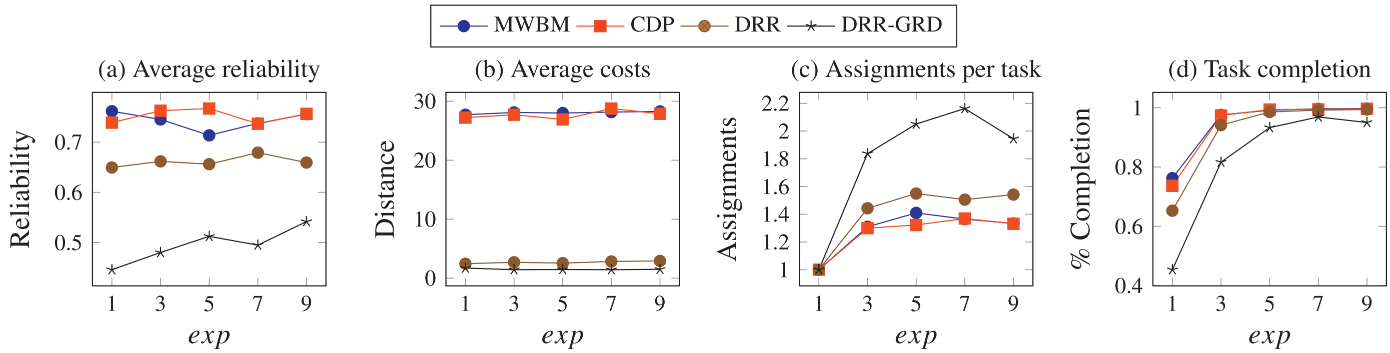

Fig. A3. Effects of task expiry time (Skewed data).

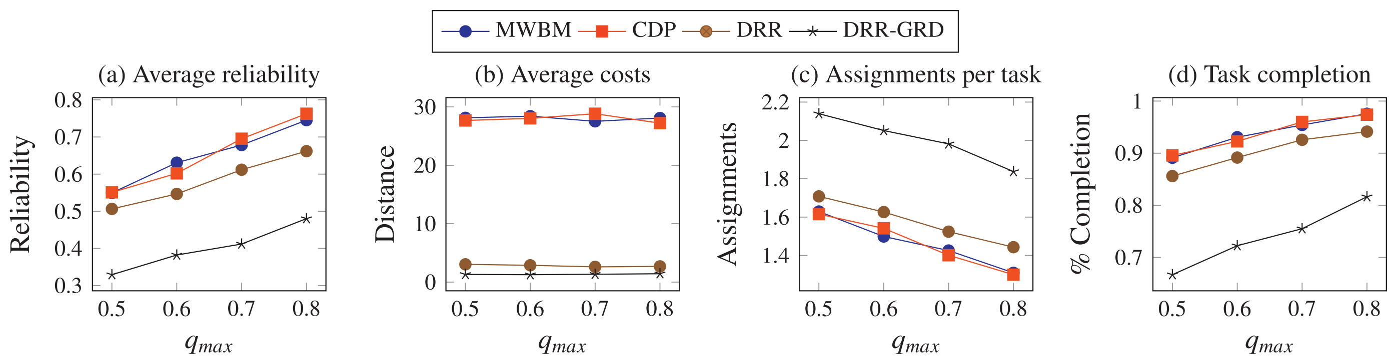

Fig. A4. Effects of the range of worker reliability (Skewed data).

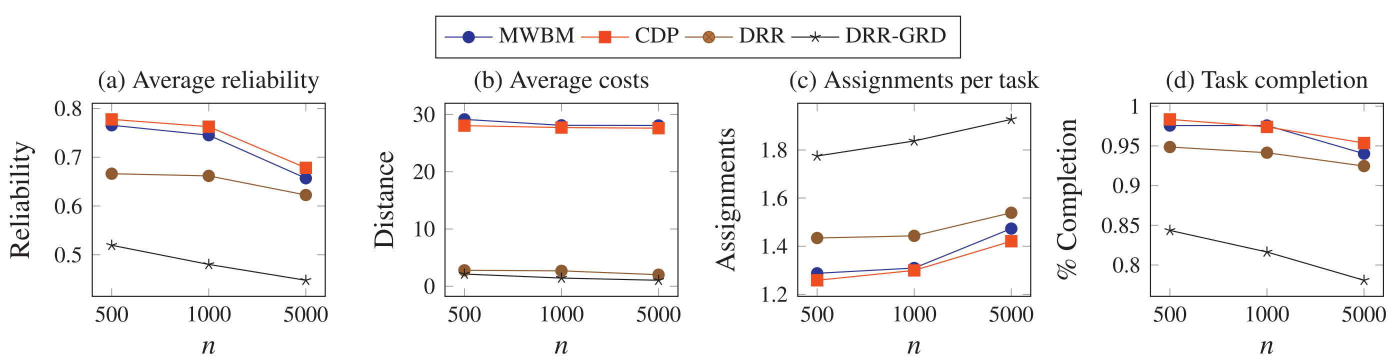

Fig. A5. Effects of the number of tasks (Skewed data).

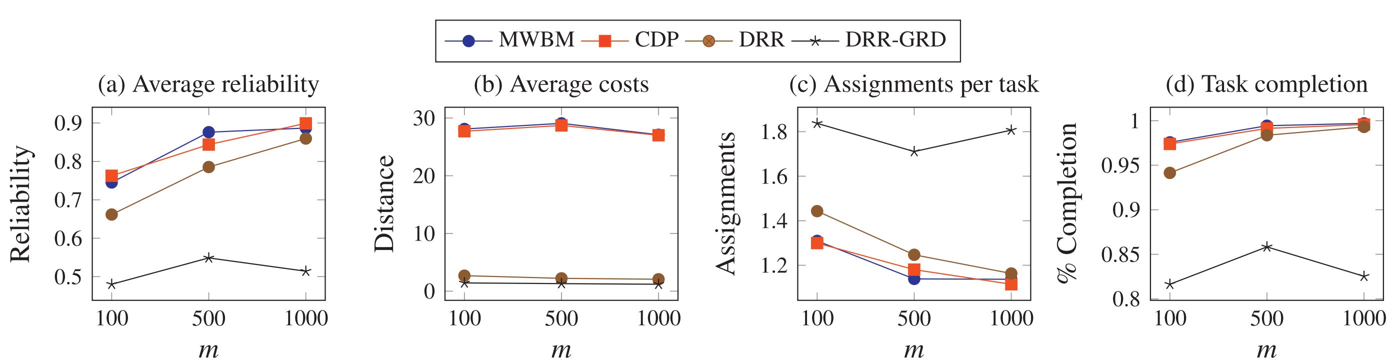

Fig. A6. Effects of the range of the number of workers (Skewed data).

degraded relative to the results of the Uniform distributions. Otherwise, the patterns of algorithmic performance remains the same. Fig. A3d the percentage of tasks completed for all algorithms. Apart for DRR-GRD, all algorithms reach new maximum completion rate with more that three rounds of expiry time.

Effects of workers' reliability

Fig. A4 shows the effects of the range of worker reliability on the comparative performance of algorithms. The reliability increase linearly with the increase in the range of worker reliabilities, for all workers (Fig. A4a). The DRR based algorithms consistently achieve small travel costs with no effects due to the changes in worker reliabilities (Fig. A4b). The relative performance, in terms of the number of assignments, decrease constantly for all algorithms (Fig. A4c). Similar, to the task reliability the percentage of completed tasks increases with increase in worker reliability range (Fig. A4d).

Effects of number of tasks and workers

Fig. A5 shows the effects of varying the number tasks with parameter n; therefore, there are more tasks available for assignment during each round. The task reliability decreases with increasing number tasks (Fig. A5a), where MWBM and CDP approach show highest decrease for $n = 5000$. The average travel costs still remains the same even when there are ten times more tasks during a round (Fig. A5b). The performance of algorithms in terms of the number of assignments per task is the opposite of the reliability performance (Fig. A5c).

Fig. A6 shows the effects of varying the number of workers with parameter m; such that, there are more alternatives available for assignment during each round. The average reliability of tasks increases due to more worker being assigned with higher q_j, as the number of workers increase (Fig. A6a). The number of workers does not have significant effect of the percentage completion of tasks (Fig. A6d).

References

Abraham, I., Alonso, O., Kandylas, V., & Slivkins, A. (2013). Adaptive crowdsourcing algorithms for the bandit survey problem. In *Conference on learning theory* (pp. 882–910).

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network flows: Theory, algorithms, and applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Audibert, J.-Y., Bubeck, S., & Lugosi, G. (2013). Regret in online combinatorial optimization. *Mathematics of Operations Research, 39*, 31–45.

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002a). Finite-time analysis of the multiarmed bandit problem. *Machine learning, 47*, 235–256.

Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. (2002b). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing, 32*, 48–77.

Bajalinov, E. B. (2003). *Linear-fractional programming: Theory, methods, applications and software*: Vol. 84. Boston, USA: Springer.

Barto, A. G. (1998). *Reinforcement learning: An introduction*. Boston: MIT press.

Berry, D. A., & Fristedt, B. (1985). *Bandit problems: Sequential allocation of experiments (monographs on statistics and applied probability)*. London: Springer.

Bubeck, S., & Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning, 5*, 1–122.

Burkard, R. E., Dell'Amico, M., & Martello, S. (2009). *Assignment problems, revised reprint*. Philadelphia: Siam.

Charnes, A., & Cooper, W. W. (1962). Programming with linear fractional functionals. *Naval research logistics quarterly, 9*, 181–186.

Chen, C., Cheng, S.-F., Gunawan, A., Misra, A., Dasgupta, K., & Chander, D. (2014). Traccs: Trajectory-aware coordinated urban crowd-sourcing. In *Proceedings of the 2nd AAAI conference on human computation and crowdsourcing*.

Chen, C., Cheng, S.-F., Lau, H. C., & Misra, A. (2015). Towards city-scale mobile crowdsourcing: Task recommendations under trajectory uncertainties. In *International joint conferences on artificial intelligence*.

Chen, W., Wang, Y., & Yuan, Y. (2013a). Combinatorial multi-armed bandit: General framework and applications. In *Proceedings of the 30th international conference on machine learning* (pp. 151–159).

Chen, X., Lin, Q., & Zhou, D. (2013b). Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *Proceedings of the 30th international conference on machine learning (ICML-13)* (pp. 64–72).

Cheng, P., Lian, X., Chen, Z., Chen, L., Han, J., & Zhao, J. (2015). Reliable diversity-based spatial crowdsourcing by moving workers. *Proceedings of the VLDB endowment, 8*.

Dekel, O., Gentile, C., & Sridharan, K. (2012). Selective sampling and active learning from single and multiple teachers. *The Journal of Machine Learning Research, 13*, 2655–2697.

Deng, D., Shahabi, C., & Demiryurek, U. (2013). Maximizing the number of worker's self-selected tasks in spatial crowdsourcing. In *Proceedings of the 21st ACM SIGSPATIAL international conference on advances in geographic information systems* (pp. 314–323). ACM.

Difallah, D. E., Demartini, G., & Cudré-Mauroux, P. (2013). Pick-a-crowd: Tell me what you like, and i'll tell you what to do. In *Proceedings of the 22nd international conference on world wide web* (pp. 367–374).International World Wide Web Conferences Steering Committee

Dinkelbach, W. (1967). On nonlinear fractional programming. *Management Science, 13*, 492–498.

Feng, Z., Zhu, Y., Zhang, Q., Ni, L. M., & Vasilakos, A. V. (2014). Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing. In *Proceedings of the IEEE INFOCOM conference* (pp. 1231–1239). IEEE.

Ganti, R. K., Ye, F., & Lei, H. (2011). Mobile crowdsensing: Current state and future challenges. *Communications Magazine, IEEE, 49*, 32–39.

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

Gittins, J., Glazebrook, K., & Weber, R. (2011). *Multi-armed bandit allocation indices*. New York: John Wiley & Sons.

Gittins, J. C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological), 41*, 148–177.

Hart, W. E., Laird, C., Watson, J.-P., & Woodruff, D. L. (2012). *Pyomo–optimization modeling in python*: Vol. 67. Berlin: Springer Science & Business Media.

Hassan, U. U., & Curry, E. (2013). A capability requirements approach for predicting worker performance in crowdsourcing. In *Proceedings of the 11th international conference on collaborative computing: Networking, applications and worksharing* (pp. 429–437). IEEE.

Hassan, U. U., & Curry, E. (2014). A multi-armed bandit approach to online spatial task assignment. In *Proceedings of the 11th IEEE international conference onubiquitous intelligence and computing*. IEEE.

Hassan, U. U., O'Riain, S., & Curry, E. (2013). Effects of expertise assessment on the quality of task routing in human computation. In *Proceedings of the 2nd international workshop on social media for crowdsourcing and human computation*. BCS.

Ho, C.-J., Jabbari, S., & Vaughan, J. W. (2013). Adaptive task assignment for crowdsourced classification. In *Proceedings of the 30th international conference on machine learning (ICML-13)* (pp. 534–542).

Ho, C.-J., & Vaughan, J. W. (2012). Online task assignment in crowdsourcing markets.. In *AAAI conference on artificial intelligence*.

Horton, J. J., & Chilton, L. B. (2010). The labor economics of paid crowdsourcing. In *Proceedings of the 11th ACM conference on electronic commerce* (pp. 209–218). ACM.

Jonker, R., & Volgenant, A. (1987). A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing, 38*, 325–340.

Karger, D. R., Oh, S., & Shah, D. (2011a). Budget-optimal crowdsourcing using low-rank matrix approximations. In *Communication, control, and computing (allerton), 2011 49th annual allerton conference on* (pp. 284–291). IEEE.

Karger, D. R., Oh, S., & Shah, D. (2011b). Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems* (pp. 1953–1961).

Kazemi, L., & Shahabi, C. (2012). Geocrowd: Enabling query answering with spatial crowdsourcing. In *Proceedings of the 20th international conference on advances in geographic information systems* (pp. 189–198). ACM.

Kittur, A., Nickerson, J. V., Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J., et al. (2013). The future of crowd work. In *Proceedings of the 2013 conference on computer supported cooperative work* (pp. 1301–1318). ACM.

Kulkarni, A., Gutheim, P., Narula, P., Rolnitzky, D., Parikh, T., & Hartmann, B. (2012). Mobileworks: Designing for quality in a managed crowdsourcing architecture. *Internet Computing, IEEE, 16*, 28–35.

Law, E., & Ahn, L. v. (2011). Human computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning, 5*, 1–121.

Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on world wide web* (pp. 661–670). ACM.

Li, L., Chu, W., Langford, J., & Wang, X. (2011). Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on web search and data mining* (pp. 297–306). ACM.

Megiddo, N. (1979). Combinatorial optimization with rational objective functions. *Mathematics of Operations Research, 4*, 414–424.

Musthag, M., & Ganesan, D. (2013). Labor dynamics in a mobile micro-task market. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 641–650). ACM.

Ong, S. P., Richards, W. D., Jain, A., Hautier, G., Kocher, M., Cholia, S., et al. (2013). Python materials genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science, 68*, 314–319.

Pentico, D. W. (2007). Assignment problems: A golden anniversary survey. *European Journal of Operational Research, 176*, 774–793.

Powell, W. B. (2007). *Approximate dynamic programming: Solving the curses of dimensionality*: Vol. 703. New York: John Wiley & Sons.

Radzik, T. (1992). Newton's method for fractional combinatorial optimization. In *Proceedings of the 33rd annual symposium on foundations of computer science* (pp. 659–669). IEEE.
Schall, D. (2012). *Service-oriented crowdsourcing: Architecture, protocols and algorithms*. New York, NY, USA: Springer Science & Business Media.
Scott, S. L. (2010). A modern bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry, 26*, 639–658.
Shalev-Shwartz, S. (2011). Online learning and online convex optimization. *Foundations and Trends in Machine Learning, 4*, 107–194.
Slivkins, A., & Vaughan, J. W. (2013). Online decision making in crowdsourcing markets: Theoretical challenges. *ACM SIGecom Exchanges, 12*, 4–23.
Tarasov, A., Delany, S. J., & Mac Namee, B. (2014). Dynamic estimation of worker reliability in crowdsourcing for regression tasks: Making it work. *Expert Systems with Applications, 41*, 6190–6210.
Teodoro, R., Ozturk, P., Naaman, M., Mason, W., & Lindqvist, J. (2014). The motivations and experiences of the on-demand mobile workforce. In *Proceedings of the 17th ACM conference on computer supported cooperative work & social computing* (pp. 236–247). ACM.
Thebault-Spieker, J., Terveen, L., & Hecht, B. (2015). Avoiding the south side and the suburbs: The geography of mobile crowdsourcing markets. In *18th ACM conference on computer-supported cooperative work and social computing*.
To, H., Ghinita, G., & Shahabi, C. (2014). A framework for protecting worker location privacy in spatial crowdsourcing. *Proceedings of the VLDB Endowment, 7*(10), 919–930.
To, H., Shahabi, C., & Kazemi, L. (2015). A server-assigned spatial crowdsourcing framework. *ACM Tranactions on Spatial Algorithms System, 1*(1), 2:1–2:28.
Tran-Thanh, L., Stein, S., Rogers, A., & Jennings, N. R. (2014). Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artificial Intelligence, 214*, 89–111.
Vermorel, J., & Mohri, M. (2005). Multi-armed bandit algorithms and empirical evaluation. In *Proceedings of the 16th European conference on Machine Learning* (pp. 437–448). Springer-Verlag: Springer.
Xiao, M., Wu, J., Huang, L., Wang, Y., & Liu, C. (2015). Multi-task assignment for crowdsensing in mobile social networks. In *Proceedings of the IEEE INFOCOM conference*. IEEE.
Yang, D., Zhang, D., Zheng, V. W., & Yu, Z. (2015). Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on, 45*, 129–142.
Zhang, D., Wang, L., Xiong, H., & Guo, B. (2014). 4w1h in mobile crowdsensing. *Communications Magazine, IEEE, 52*, 42–48.
Zou, G., Gil, A., & Tharayil, M. (2014). An agent-based model for crowdsourcing systems. In *Proceedings of the 2014 winter simulation conference* (pp. 407–418). IEEE Press.

Umair ul Hassan is a PhD candidate at the Insight Centre of Data Analytics, National University of Ireland, Galway. His research tackles the adaptive assignment problems in crowdsourcing. He investigates the problem in the data curation, energy management, and spatial crowdsourcing domains. His current projects include the design and implementation of the task management middleware for collaborative data management.

Edward Curry is a research leader at the Insight Centre for Data Analytics (www.insight-centre.org) and a funded investigator at LERO the Irish Software Research Centre (www.lero.ie). Edward has worked extensively with industry and government advising on the adoption patterns, practicalities, and benefits of new technologies. Edward has published over 120 scientific articles in journals, books, and international conferences. He has presented at numerous events and has given invited talks at Berkeley, Stanford, and MIT. In 2010, he was a guest speaker at the MIT Sloan CIO Symposium to an audience of 600+ CIOs and senior IT executives. His research projects include studies of smart cities, energy intelligence, semantic information management, event-based systems, and collaborative data management. He is a member of the scientific leadership committee of Insight, and a lecturer in Informatics at the National University of Ireland Galway (NUIG). He is the Vice President of the Big Data Value Association (www.BDVA.eu) a non-profit industry-led organization with the objective of increasing the competitiveness of European companies with data-driven innovation.