Sun and Tan *Hum. Cent. Comput. Inf. Sci.* (2019) 9:25
https://doi.org/10.1186/s13673-019-0187-4

Human-centric Computing and Information Sciences

RESEARCH Open Access

A trust-aware task allocation method using deep q-learning for uncertain mobile crowdsourcing

Yong Sun[1,2*] and Wenan Tan[3]

*Correspondence:
ysun.nuaa@yahoo.com;
syong@nuaa.edu.cn
[1] Anhui Center for Collaborative Innovation in Geographical Information Integration and Application, Chuzhou University, Chuzhou 239000, Anhui, China
Full list of author information is available at the end of the article

Abstract

Mobile crowdsourcing has emerged as a promising collaboration paradigm in which each spatial task requires a set of mobile workers in near vicinity to the target location. Considering the desired privacy of the participating mobile devices, trust is considered to be an important factor to enable effective collaboration in mobile crowdsourcing. The main impediment to the success of mobile crowdsourcing is the allocation of trustworthy mobile workers to nearby spatial tasks for collaboration. This process becomes substantially more challenging for large-scale online spatial task allocations in uncertain mobile crowdsourcing systems. The uncertainty can mislead the task allocation, resulting in performance degradation. Moreover, the large-scale nature of real-world crowdsourcing poses a considerable challenge to spatial task allocation in uncertain environments. To address the aforementioned challenges, first, an optimization problem of mobile crowdsourcing task allocation is formulated to maximize the trustworthiness of workers and minimize movement distance costs. Second, for the uncertain crowdsourcing scenario, a Markov decision process-based mobile crowdsourcing model (MCMDP) is formulated to illustrate the dynamic trust-aware task allocation problem. Third, to solve large-scale MCMDP problems in a stable manner, this study proposes an improved deep Q-learning-based trust-aware task allocation (ImprovedDQL-TTA) algorithm that combines trust-aware task allocation and deep Q-learning as an improvement over the uncertain mobile crowdsourcing systems. Finally, experimental results illustrate that the ImprovedDQL-TTA algorithm can stably converge in a number of training iterations. Compared with the reference algorithm, our proposed algorithm achieves effective solutions on the experimental data sets.

Keywords: Collaborative computing, Mobile crowdsourcing, Dynamic task allocation, Collaborative partners selection, Trust optimization

Introduction

With the advancing technology of mobile devices with numerous built-in sensors, mobile crowdsourcing has recently emerged as a new collaboration paradigm in numerous intelligent mobile information systems [1]. The existing mobile crowdsourcing has applications in numerous domains including urban planning, traffic monitoring, ride sharing, environmental monitoring and intelligent disaster response [2]. Mobile crowdsourcing is a combination of spatial crowdsourcing and smart phone technology that employs mobile workers to perform certain tasks in a specific

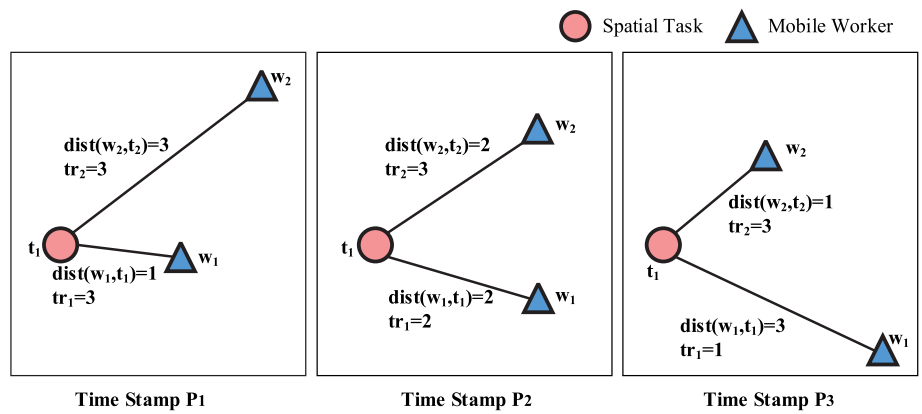

Fig. 1 A mobile crowdsourcing example

location [3]. For example, in a disaster search and rescue scenario, the requester urgently needs to collect images and videos of search areas from different locations in a country [4]. The requester submits a query to a mobile crowdsourcing server. Then, the server allocates the spatial tasks to the available workers in the vicinity of the disaster location.

Geographic information plays a key factor in many aspects of mobile activities [5]. The goal of typical mobile crowdsourcing is to allocate multiple tasks to a team of suitable workers located within their time zone [6]. The required tasks of mobile crowdsourcing are considered with strong spatial proximity optimization [2]. For the above instance, it is important to rapidly respond to emergencies or disasters [3]. Mobile crowdsourcing systems must assign emergency tasks to workers who are in the vicinity to the target location. Most tasks must be accomplished within a set time, making it impossible for mobile workers to travel long distances to accomplish the required tasks [4].

Moreover, the success of mobile crowdsourcing relies heavily on the quality of location-related workers [7]. The existing crowdsourcing systems are dependent on mainly mobile workers to allocate tasks to themselves when logging on to the systems [8], and many spatial tasks may not be allocated to suitable workers [9]. The execution quality of the crowdsourcing tasks suffers because the workers may be malicious participants [10–13]. The trustworthiness of mobile workers must be considered in the mobile crowdsourcing setting [12]. In this context, mobile crowdsourcing should consider both the trustworthiness and location of mobile workers. This paper focuses on the trust-aware task allocation (TTA) optimization problem of mobile crowdsourcing systems.

The objective of optimizing TTA is to maximize the trust score and minimize the distance cost of mobile crowdsourcing. In the real world, mobile crowdsourcing systems are inherently dynamic, and the trust scores of mobile workers are unknown. The mobile crowdsourcing scenario in Fig. 1 has a location-based task $t_i (i = 1)$, shown in red circles, and two crowd workers $w_i (1 \leq i \leq 2)$, shown as blue triangles. At the time stamp $P_i (1 \leq i \leq 3)$, worker w_i and location-based tasks t_i join the mobile crowdsourcing system. Assume that spatial tasks t_1) can be accomplished by w_1 and w_2 who

Table 1 A mobile crowdsourcing example

Time	Crowdsourcing pair	Trust	Dist	Crowdsourcing pair	Trust	Dist
P_1	(w_1, t_1)	3	1	(w_2, t_1)	1	3
P_2	(w_1, t_1)	2	2	(w_2, t_1)	2	2
P_3	(w_1, t_1)	1	3	(w_2, t_1)	3	1

have some compensation traveling distance $dist(w_i, t_j)$ and trust score tr_i, as described as Table 1. Based on our observation, at time stamp P_1, mobile worker w_1 can be recommended to do the spatial tasks t_1 because of the high trust scores and low travel cost; by contrast, at time stamp P_2 and P_3, mobile worker w_2 may be selected for task t_1.

As mentioned above, mobile workers frequently move to different locations, and trust scores for performing the required tasks are unstable in mobile crowdsourcing systems. Mobile crowdsourcing systems require optimization to be dynamic and adaptive to address this uncertainty [2]. However, mobile crowdsourcing emerged very recently and is typically considered to be a static environment in most existing research [3–5]. Most of the current crowdsourcing approaches have vital drawbacks. Mathematical optimization algorithms, in which the evaluation parameters are considered to be certain and fully known in advance, are used to solve the allocation problems of static crowdsourcing systems [2, 6].

The advantage of mobile crowdsourcing is to enable a crowd of mobile workers to offer collaboration services, which enhances the efficiency of performing cooperative tasks while reducing the cost. Unfortunately, static approaches may fail when dealing with task allocations in uncertain mobile crowdsourcing. TTA optimization algorithms should adapt to frequent changes in crowdsourcing systems. The inherently dynamic changes in mobile crowdsourcing systems are difficult to handle. An attempt has been made to design an adaptive learning algorithm to solve this uncertain problem. In [6], Q-learning is employed for the dynamic task allocation of crowdsourcing systems. However, Q-learning has mainly been limited in applicability in addressing only medium-sized optimization problems [14–16]. The majority of real-world TTA problems are fundamentally large-scale, and the crowdsourcing state space is extremely large because massive spatial tasks and mobile workers exist on mobile crowdsourcing systems. It is a considerable challenge to solve large-scale task allocation in uncertain scenarios, which further highlights the need for designing innovative and highly effective learning algorithms to optimize the real-world TTA problems.

In summary, a dynamic TTA algorithm is needed to enhance the model performance by fully exploiting the potential advantage in uncertain mobile crowdsourcing systems. The difficulty lies in accurately modeling the dynamic characteristic of task allocations and making better crowdsourcing decisions, with the aim of maximizing the model performance over a long period of time. Specifically, the dynamic TTA optimization can be formulated as a Markov decision process (MDP) problem. The emerging deep Q-learning (DQL) algorithm shows distinct advantages for large-scale MDP problems and has been widely used in dynamic sequential decision making problems [17–20]. By combining the advantages of both deep neural networks

and Q-learning, DQL is able to predict the next state and action in large-scale crowdsourcing environments.

This paper investigates a practical and important problem, namely dynamic trust-aware task allocation, which aims to maximize the trust score and minimize the travel distance cost in uncertain crowdsourcing environments. This paper mainly focuses on addressing the uncertain large-scale task allocation in location-based mobile crowdsourcing. Our proposed DQL-TTA algorithm can be directly extended to the scenario of large-scale task allocation in real-world mobile crowdsourcing. The principal contributions of our work can be summarized as below:

- TTA optimization is formally defined in mobile crowdsourcing systems. For an uncertain scenario, dynamic TTA optimization is formalized as an MDP-based mobile crowdsourcing model (MCMDP). MCMDP comprises four core elements, including the crowdsourcing state, allocation action and immediate reward.
- We first study a Q-learning algorithm to optimize the dynamic allocations in mobile crowdsourcing. In addition, deep Q-learning-based trust-aware task allocation (DQL-TTA) is proposed to handle large-scale MCMDP optimization problems, which are intractable in traditional Q-learning. Tabular Q-Learning is further advanced by deep neural networks to estimate the Q-value of the next crowdsourcing state in a practical manner. The DQL-TTA algorithm extends the TTA problems to dynamic optimization by combining the advantage of both deep Q-network and bi-objective trust optimization.
- To improve the overall performance DQL-TTA, this paper proposes an improved DQL-TTA (ImprovedDQL-TTA) algorithm to handle the large-scale MCMDP problems much more stably in real-world crowdsourcing scenarios. The novel deep neural network architecture with an action advantage function is integrated in ImprovedDQL-TTA, which performs better than the DQL-TTA algorithm in mobile crowdsourcing. The pivotal idea of ImprovedDQL-TTA is to design two estimators to separately learn the state value and action advantage functions with two streams of fully connected neural network layers. Additionally, the mini-batch stochastic gradient descent with advanced training mechanisms and an Epsilon-decreasing greedy policy are integrated into ImprovedDQL-TTA. In this context, ImprovedDQL-TTA can maintain good stability to solve large-scale trust-aware allocation problems in uncertain mobile crowdsourcing. Theoretical analysis is conducted to demonstrate the applicability of ImprovedDQL-TTA.
- The experimental results illustrate that the proposed ImprovedDQL-TTA algorithm can achieve greater effectiveness and stability in uncertain scenarios of large-scale mobile crowdsourcing systems.

The rest of this paper is organized as follows. We discuss related work on dynamic mobile crowdsourcing in "Related work" section. The preliminaries and formulation of mobile crowdsourcing are presented in "Preliminaries and problem formulation" section. In "Trust-aware task allocation with deep Q-learning" section, the improved deep Q-learning-based trust aware task allocation algorithm is proposed for uncertain mobile crowdsourcing. An experimental study conducted to illustrate the value of the

proposed algorithm is discussed in "Experimental results and analysis" section, followed by conclusions.

Related work

Crowdsourcing is a newly emerging field that enables organizations or companies to make their requests on numerous intelligent web platforms [21], such as Amazon MTurk, Upwork, and Crowdflower [22]. Crowdsourcing has been widely applied to annotations [23], graph searching [24], data analysis [25], query processing [26], and social-aware collaborations [27, 28]. In such applications, the required tasks can be accomplished by online workers on basis of crowdsourcing techniques. However, these workers do not have to travel to the target locations to accomplish the required tasks. Unlike general crowdsourcing, location-based mobile crowdsourcing systems usually require mobile workers to move to the specified location to perform tasks.

Task allocation in location-based mobile crowdsourcing

Mobile crowdsourcing entails a novel mechanism for tasks performed by mobile workers. Task allocation in the location-based mobile crowdsourcing has gained increasing attention in recent years [29, 30]. Location-based mobile crowdsourcing is a subclass of spatial crowdsourcing that allocates available mobile workers to spatial tasks on a mobile crowdsourcing system [2]. A task allocation framework was formally presented for the location-based mobile crowdsourcing [3]. Kazemi and Shahabi proposed a task assignment problem for spatial crowdsourcing [12]. They proposed a network-flow-based algorithm for handling the allocation problem. The goal of this framework is to maximize the number of tasks matched with workers [31]. To extended this spatial allocation problem to the maximum score assignment problem for skills-based crowdsourcing [3]. To handle the large-scale query problem, Li et al. proposed R-Tree-based approximation algorithms for task allocation in mobile crowdsourcing [5]. Recently, an optimal task allocation problem was presented to address the quality constraints [29].

Considering the private participating mobile devices [8], Tran and To et al. proposed a real-time algorithm for spatial task allocation in server-assigned crowdsourcing [5]. This framework can be employed to protect the real locations of mobile workers and to maximize the crowdsourcing success rates [10, 11]. Unlike private location-based queries, this study focuses on trust-aware task allocation in mobile crowdsourcing. The workers in crowdsourcing processing are not always trusted [13]; thus, another work by Kazemi aimed to address the optimization of trust in task allocation [12]. The trustworthiness of mobile workers must be considered in the mobile crowdsourcing setting [12]. The goal is to optimize spatial proximity and trustworthiness management in task allocation. For participatory mobile systems, trust evaluation is an effective mechanism to promote mobile system performance by identifying the trustworthiness of potential participants [13]. This process can formally be defined as processing spatial tasks for a crowd of trustworthy mobile workers close to the target location.

Uncertain mobile crowdsourcing and deep reinforcement learning

In the existing research, mobile crowdsourcing is considered to occur in a stationary environment, in which the crowdsourcing quality is considered to be a certain parameter

that is fully known in advance [2]. Crowdsourcing is defined as a static problem, and the multiple quality objectives are invariant [2]. By contrast, in uncertain crowdsourcing systems, all parameter values may change spontaneously [6]. Thus, Cheng et al. proposed a prediction-based allocation strategy to estimate the location and quality distributions of future workers and tasks for global optimal task allocation [32]. In comparison, our proposed method aims to maximize the reward utility on the basis of deep Q-learning algorithms by adapting the dynamic trust-aware allocation strategy to the uncertain mobile crowdsourcing environment.

Q-learning algorithms have been found to be more suitable for uncertain problems of crowdsourcing [2, 6]. A Q-Learning agent learns how to address uncertain decision-making problems with dynamic environments [14]. Since tabular-Q learning requires iterative updating to converge, an optimal policy is difficult to find in limited time [14–16]. To solve this problem, deep Q-networks algorithm was proposed by combining Q-learning with deep neural networks [17–20]. Van Hasselt, Guez and Silver proposed double DQN to address the overestimation of Q-learning [17]. Schaulman et al. designed a prioritized experience replay mechanism to enhance the efficiency of training data [33]. Wang et al. proposed dueling DQN to further improve the convergence speed [34]. The dueling DQN algorithm represents both the state value function and the related advantage function [35].

A Q-learning algorithm was adopted in our previous work to obtain the optimal allocation policy in uncertain crowdsourcing [6]. However, the Q-learning algorithm is limited by its slow convergence in the large crowdsourcing state and action space. To address this limitation, we propose a novel neural network with advanced deep Q-learning algorithm. This algorithm extends the TTA optimization to dynamic crowdsourcing problems by means of a deep Q-learning algorithm. Most importantly, we propose an improved adaptive optimization algorithm by combining TTA optimization and advanced deep Q-learning, which maintains great efficiency in solving large-scale TTA problems in uncertain mobile crowdsourcing.

Preliminaries and problem formulation

In this section, a trust-aware task allocation scenario is formally introduced to deal address challenges in unreliable mobile crowdsourcing environments.

Mobile crowdsourcing preliminaries

The basic concepts of mobile crowdsourcing systems [2, 3, 6] are formally defined as follows.

Definition 1 (*Spatial task*) Denote a spatial task st as a tuple: $st = \langle expir, loca, stype \rangle$. The textual property describes the task submitted by the requester. The location property *loca* denotes the location coordinates in relation to the required task. The expiry property *expir* is the specific time of task completion. The type property *stype* indicates the spatial task type.

A task st can be accomplished by a mobile worker only if the mobile worker physically travels to the target location *loc* within the specific time *expired*. All spatial tasks have

time constraints, and the mobile workers must physically travel to the target location on time. Mobile workers are formulated as follows.

Definition 2 (*Mobile worker*) Denote a mobile worker as a tuple: $cw = \langle hisinfo, exptype, loc \rangle$. The property *hisinfo* is a crowdsourcing data sequence that records the series of spatial tasks allocated to mobile worker *cw*. A mobile worker cw_i has his own expertise *exptype* to be competent for a type *stype* of crowdsourcing task st_i. The competence of worker cw_t for task st_t can be evaluated in terms of a quality score. The location *loc* represents the current location of the mobile worker.

A worker cw_t is associated with the traveling cost $dist(cw_t, st_j)$, and $dist(cw_t, st_j)$ is the traveling distance between cw_t and st_j. Accordingly, mobile workers cw_t are recommended to perform spatial tasks st_j if they are in the near vicinity of mobile worker cw_t.

Definition 3 (*Travel distance*) Distance $f_{dist}(x_i^j)$ specifies the travel cost in terms of the movement required to get from the location *aloc* of mobile worker cw_j to the location *bloc* of spatial task st_i. The distance may be computed on the basis of the Euclidean distance metric.

$$f_{dist}(x_i^j) = \sqrt{(aloc_x - bloc_x)^2 + (aloc_y - bloc_y)^2} \tag{1}$$

where $(aloc_x, aloc_y)$ and $(bloc_x, bloc_y)$ are the coordinates of *aloc* and *bloc*.

In the optimization process, the algorithms wish to allocate workers cw_t to spatial tasks t_t with a minimum traveling cost so that the sum quality value of the allocation is maximized and the total distance cost is minimized [6]. However, in uncertain environments, numerous discrete events cause the execution failure of spatial tasks. Therefore, a trust-aware allocation optimization metric is required for solving the unreliable quality problem of mobile crowdsourcing systems.

Trust assessment metric

The main impediment to the success of spatial task allocation is the issue of trust evaluation for mobile workers. To evaluate the trustworthiness of a mobile worker, we consider and evaluate two parameters: *reputation* and *expertise*. The calculation of each trust parameter is discussed first; then, the trust evaluation is explained in detail.

The reputation of a worker reflects the probability, calculated based on historical data, of completing a spatial task. In general, the reputation of mobile workers can be described with reference to their mobile worker IDs and trust parameters. The reputation metric of a worker is formally represented by definition 4.

Definition 4 (*Worker reputation*) The reputation of a mobile worker is denoted as $wqos = \langle id_{cw}, rep \rangle$, where id_{cw} represents the mobile worker ID and *rep* is the reputation value of the mobile worker. rep_t denotes the reputation attribute of the i-th mobile worker *cw*. The i-th mobile worker is determined by observation of whether the previous

interactions among mobile workers result in successful task execution. The observation is often described by two variables: n_i, denoting the number of successful interactions, and N_i, denoting the total number of interactions for the i-th mobile worker. The trust value can be calculated as:

$$rep_t = \frac{n_i + 1}{N_i + 2} \tag{2}$$

where the trust value of a service is initialized to 1/2.

Definition 5 (*Worker expertise*) Denote expertise as the knowledge estimation of a mobile worker, which is especially important for spatial tasks that require particular knowledge, such as geomatics skills and familiarity with geographical information science. Denote the matching between the i-th mobile worker's skills and j-th task requirements by E_i^j. Suppose that E_j^{ta} is the requirements for the i-th task and that E_i^{cw} is the collection of expertise of mobile worker cw_t; then,

$$E_i^j = \frac{|E_i^{ta} \cap E_j^{cw}|}{|E_i^{ta}|} \tag{3}$$

where the trust value of a service is initialized to 1/2.

All the trust parameters are combined into a single value for computing the trust score of the mobile worker.

$$f_{tr}(x_i^j) = w_{rep} \cdot rep_t + w_{exp} \cdot E_i^j \tag{4}$$

where w_{rep} and w_{exp} are the weights of each crowdsourcing parameter, $w_{rep} + w_{exp} = 1$.

In reality, task allocation is not a static decision process, as spatial tasks and mobile workers interact dynamically with the system. The allocation decision process is conducted iteratively, and each iteration involves allocating spatial tasks to trustworthy mobile workers in uncertain scenarios.

Weighting TTA approach with normalization

TTA problems focus not only on trust management [36] but also spatial optimization. The aim of a crowdsourcing problem is to match tasks and mobile workers such that the trust score is maximized and the allocation distance is minimized. The objective functions of TTA are normalized between 0 and 1, and the bi-objective allocation optimization problem is formulated as follows:

$$minimize: \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{2} w_k \cdot (f_k(x_i^j) - z_k^U)/(z_k^N - z_k^U) \tag{5}$$

$$\begin{aligned} \textit{subject to}: & \sum_{j=1}^{m} x_i^j = 1, \quad i = 1, ..., n \\ & \sum_{i=1}^{n} x_i^j = 1, \quad j = 1, ..., m \\ & x_i^j \in \{0, 1\}, i = 1, ..., n, j = 1, ..., m \end{aligned} \tag{6}$$

for $\sum_{k=1}^{2} w_k = 1, w_t > 0$. The minimum travel cost maximum trust-aware allocation optimization is changed to a weighted sum single objective problem with the max-min operator [36], where z_k^U is the minimum value of the k-th objective and z_k^N is the maximum value of the k-th objective; and all objective function $f_k(x)$ and $X = (x_{1,1}, ..., x_{m,n})$ is the matrix of decision variables. To solve this problem, a trust-distance weighted function is defined as an integrated optimization of trust scores and allocation distance costs. Owing to the dynamic nature of uncertain crowdsourcing scenarios, the trust values of mobile workers and tasks cannot be known in advance. In addition, many workers may be unavailable on the mobile crowdsourcing system at run time.

Trust-aware task allocation with deep Q-learning

The majority of TTA optimization approaches require prior knowledge, but such approaches are not applicable in dynamic mobile crowdsourcing environments, where the availability of mobile workers is subject to frequent and unpredictable changes [2, 6]. Let us consider the submission of spatial tasks from requesters through a mobile crowdsourcing system, whereby spatial tasks are reached in an online manner. In such a scenario, the mobile crowdsourcing system possesses no prior information regarding spatial tasks and mobile workers.

The crowdsourcing TTA optimization problem is modeled as a Markov decision process-based mobile crowdsourcing (MCMDP) problem. Deep Q-learning is introduced to address the MCMDP problem. Furthermore, we propose an improved deep Q-learning-based trust-aware task allocation (ImprovedDQL-TTA) algorithm by combining trust crowdsourcing optimization and deep Q-learning, which enables the learning agent to solve large-scale MCMDP problems in an uncertain scenario.

MDP model for uncertain mobile crowdsourcing

To address the dynamic problems of uncertain crowdsourcing TTA, a Markov decision process is adopted. The Markov decision process, a machine learning model, is a typical intelligence framework for modeling sequential decision-making problems under uncertainty [15]. In this paper, the MDP is applied to demonstrate the trust-aware task allocations and adaptation processes schematically in uncertain mobile crowdsourcing.

A mobile crowdsourcing MDP consists of a five-tuple $= \langle S, A, P, R, O \rangle$, where S is a state space composed of a finite set of crowdsourcing states, A is a crowdsourcing action space composed of a finite set of actions, P is the transition function for reaching the

next crowdsourcing state s' from state s when an action $a \in A(s)$ is performed by a crowdsourcing agent, R is a real crowdsourcing valued reward function, where the agent receives an immediate reward $r = R(s'|s, a)$, and O is the crowdsourcing observation space in which the agent can fully observe the mobile crowdsourcing decision environment. On this basis, the mobile crowdsourcing MDP can be defined as follows.

Definition 6 (*Mobile Crowdsourcing MDP (MCMDP)*) A MCMDP is formally defined as a seven-tuple: MCMDP= $\langle S^i, s_0^i, s_r^i, A^i, P^i, R^i, O^i \rangle$, where:

- S^i is the set of tasks in the state space of a particular crowdsourcing partially observed by agent i.
- $s_0^i \in S$ is the initial task and any execution of the mobile crowdsourcing beginning from this task.
- $s_r^i \in S$ represents the terminal task. When arriving at the terminal task, an execution of mobile crowdsourcing is terminated.
- A^i is the set of mobile workers that can perform tasks $s \in S^i$, and mobile worker cw belongs to A^i only if the precondition is satisfied by s.
- P is a probability value, that is, a transition distribution $P(s'|s, a)$ that determines the probability of reaching the next state s' from state s if action $a \in A(s)$ is fulfilled by a crowdsourcing agent. The probability distribution $P(s'|s, a)$ can be defined as

$$\sum_{s' \in S} P(s'|s, a) = 1, \forall s \in S, \forall a \in A. \tag{7}$$

- R^i is the reward function when mobile worker $cw \in A^i$ is invoked, agent i transits from s to s', and the learning agent obtains an immediate reward r^i. The expected value is $R^i(s'|s, ws)$. Consider selecting mobile worker cw with multiple quality criteria, where agent i receives the following quality vector as a reward:

$$QoS(s, cw, s') = [f_{tr}(s, cw, s'), f_{dist}(s, cw, s')]^T, \tag{8}$$

  where each $f_k(\cdot)$ denotes a quality attribute of mobile worker cw.
- O is the crowdsourcing observation space in which the agent can fully observe the mobile crowdsourcing decision environment.

The MCMDP solution is a collection of TTA decision policies, each of which can be described as a procedure of trust-aware task allocation $cw \in A$ by agent i in each state s. These policies, denoted as π, actually map spatial tasks to mobile workers, defined as $\pi = S \rightarrow A$. The MCMDP policy can be defined as a mobile crowdsourcing model. The main idea is to identify the optimal policy for trust-aware allocation in uncertain mobile crowdsourcing.

Deep Q-learning-based trust-aware task allocation algorithm

The above section analyzed the optimization problem of trust aware allocation by means of the MCMDP model. The optimization objective is to maximize the long-term rewards

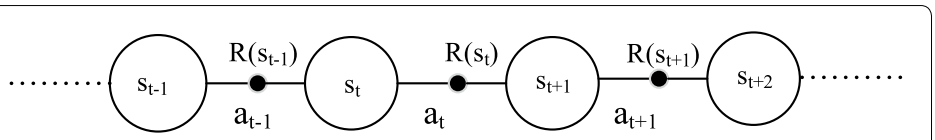

Fig. 2 A sequence of state and state-action reward pairs

of the MCMDP. The solution of the MCMDP can be denoted as a policy π that guides a learning agent to take the right action for the specific crowdsourcing state.

Dynamic task allocation with Q-Learning

The uncertain mobile crowdsourcing problem can be formulated as an MCMDP model. However, the transition probabilities are not known, and we do not initially know the rewards of taking the allocation action. In this case, Q-learning is suggested for a crowdsourcing agent to determine the optimal policy. Q-learning is a temporal difference learning algorithm [14, 15] that takes into account the fact that the agent initially has only partial knowledge of the crowdsourcing MCMDP. In general, assume that an agent learns from experience to address uncertain mobile crowdsourcing. The agent can obtain a set of state-action rewards $\langle s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_t, a_t, r_t \rangle$, which indicates that the agent was in state s_t, selected action a_t, and obtained reward r_t. Figure 2 illustrates the sequence of the crowdsourcing state and state-action reward pairs.

Temporal difference learning agents determine the increment to $V(s_t)$ in each time step. At time t, the agents immediately create an update by using discount rewards and computing $V(s_t)$. Temporal difference learning [15] can defined as

$$V(s_t) = V(s_{t-1}) + \alpha \cdot \big(r_t + \gamma \cdot V(s_t) - V(s_{t-1})\big). \tag{9}$$

The goal of temporal difference learning agents is to update $V(s_t)$ by $R(s_t) + \gamma \cdot V(s_t)$ in each step. Tabular Q-learning is a common approach in temporal difference learning for maximizing total rewards. For each state s and action a, the tabular Q-learning algorithm takes an action, observes a reward r, enters a next state s', and updates $Q(s, a)$. The key of the Q-learning algorithm is a straightforward value $Q(s, a)$ iteration update. $Q(s, a)$ is accumulated for the current estimate of Q^{π} in each training iteration. The learning table values of $Q(s, a)$ are revised by the following function:

$$Q^{\pi}(s, a) = (1 - \alpha) \cdot Q(s, a) + \alpha \cdot \Big(r + \gamma \cdot \max_{a'} Q(s', a')\Big). \tag{10}$$

The learning rate $\alpha \in [0, 1]$ indicates the extent to which the existing estimation of $Q^{\pi}(s, a)$ contributes to the next estimation. The $Q(s, a)$ values ultimately converge to the optimum value $Q^{*}(s, a)$ [15]. Thus, the Q-learning-based allocation algorithm ultimately discovers an optimal policy for any finite MCMDP [6]. The basic optimization involves incorporating both the travel distance and the trust score of mobile workers into the dynamic mobile crowdsourcing decisions. Thus, the reward function of Q Learning-based TTA is defined as in Definition 7.

Definition 7 (*Reward function*) Suppose that a mobile worker completing a task can be estimated by a trust score $f_{tr}(x_i^j) = tr(x_i^j)$. Each mobile worker is required to move from location *aloc* to *bloc* when completing the spatial task, which incurs a distance cost $f_{dist}(x_i^j)$. The distance cost is evaluated in terms of the distance $f_{dist}(x_i^j) = dist(aloc, bloc)$ between *aloc* and *bloc*. As a result, the reward function is determined with *QoS* vectors $[f_{tr}(x_i^j), f_{dist}(x_i^j)]$. Owing to the different scale of each *QoS* objective, the *QoS* value is mapped into the interval [0, 1]. With the min-max operator, the learning reward function adopts the linearly weighted sum approach to calculate the value of all *QoS* objectives:

$$r = \sum_{k=1}^{2} w_k \cdot (f_k(x_i^j) - z_k^U)/(z_k^N - z_k^U) \tag{11}$$

In the training iterations, the learning agent estimates its optimal policy by maximizing the total of received crowdsourcing rewards in the uncertain scenario.

Dynamic task allocation with deep Q-learning

Tabular Q-learning is not a feasible solution owing to the large-scale state and action spaces in uncertain mobile crowdsourcing systems. Moreover, a Q-learning table is environment-specific and not generalized. In large-scale uncertain systems, there are too many states and actions to store in machine memory, and learning the value of each state is a slow process. This section introduces a new and highly effective Q-Learning-based task allocation mechanism.

To adapt to changes in large-scale mobile crowdsourcing systems, we propose a deep Q-learning-based trust-aware task allocation (DQL-TTA) algorithm that is a combination of advances in deep neural network and Q-learning techniques. Specifically, the dynamic TTA problem is formalized as a Markov decision process-based mobile crowdsourcing model. The experience of a crowdsourcing state transition is denoted as s, a, r, s', and a set of crowdsourcing states and allocation actions with a transition policy constitute an MCMDP. One episode of an MCMDP forms a limited sequence of crowdsourcing states, allocation actions and rewards:

$$s_0, a_0, r_0, S_1, a_1, r_1, s_2, ..., s_t, a_t, s_{t+1}, ..., s_{n-1}, a_{n-1}, r_{n-1}, s_n \tag{12}$$

where s_t denotes the current state, a_t denotes the current action, r_t denotes the reward after performing an action, and s_{t+1} denotes the next state in the dynamic mobile crowdsourcing system.

The DQL-TTA algorithm directly combines a deep neural network and Q-Learning to solve the dynamic trust-aware allocation problem. The DQL-TTA learning algorithm uses a value iteration approach, in which the crowdsourcing value function $Q = Q(s, a; \theta)$ is a parameterized function with parameter θ that takes crowdsourcing state S and crowdsourcing action space A as inputs and returns a crowdsourcing Q value for each action $a \in A$. Then, we can use a greedy approach to select a crowdsourcing action:

$$Q(s) = argmax_{a \in A} Q(s, a; \theta) \tag{13}$$

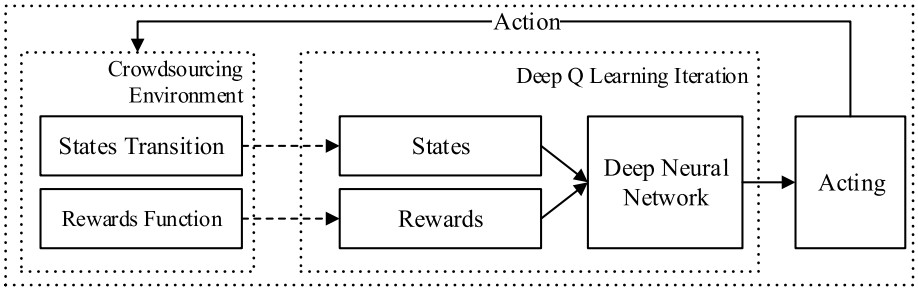

Fig. 3 Deep Q-learning-based task allocation framework

DQL-TTA iteratively solves the mobile crowdsourcing MDP problem by learning the weights of the deep neural network towards the optimization objective. The DQL-TTA algorithm differs from Q-Learning in two ways. Traditional Q-Learning is based on the Bellman equation, and the Q value is iteratively updated: $Q_{t+1}(s,a) = E[r + \gamma \cdot max_{a'} Q_t(s',a')|s,a]$. Q-Learning algorithms with value iterations are impractical for large-scale crowdsourcing problems. Thus, it is practical to employ a dynamic crowdsourcing function approximation to assess the action value function $Q(s,a;\theta) \approx Q^*(s,a)$, which is a typical function approximation.

DQL-TTA is designed as a function approximation with weight θ for the mobile crowdsourcing MDP problem. The parameters of the DQL-TTA function approximation can be learned by minimizing loss function $L(\theta_t)$, which is optimized at iteration i

$$L(\theta_t) = \mathbb{E}_\pi \left[\left(y_t - Q(s,a;\theta_t) \right)^2 \right] \tag{14}$$

where y_t is the target value for iteration i and can be computed as

$$y_t = \begin{cases} r_t, & if\ A(s') = \varnothing \\ r_t + \gamma \cdot max_{a'} Q(s',a';\theta_{t-1}), & else \end{cases} \tag{15}$$

DQL-TTA considers the crowdsourcing states and allocation actions as the inputs of a deep Q-network and outputs the Q-value for dynamic allocations. Figure 3 illustrates the deep Q-learning-based trust-aware task allocation (DQL-TTA) algorithm framework.

Dynamic task allocation with improved deep Q-learning

As discussed in [17, 33–35], the performance of deep Q-learning algorithms may not to be stable. To improve the overall performance of DQL-TTA, an improved DQL-TTA algorithm (ImprovedDQL-TTA) is further proposed to handle large-scale MCMDP problems much more stably in uncertain mobile crowdsourcing environments. Our proposed ImprovedDQL-TTA algorithm has been improved with the following important mechanisms: (i) mini-batch stochastic gradient descent approach with advanced training mechanisms; (ii) Epsilon-decreasing greedy policy; iii) a novel deep neural network architecture with an action advantage function.

Mini-batch stochastic gradient descent The parameters of ImprovedDQL-TTA from an earlier training iteration θ_{t-1} are fixed while optimizing the loss function $L(\theta_t)$. Note that the targets rely on the ImprovedDQL-TTA weight parameters. A local minimum of the loss function by the gradient is obtained as follows,

$$\begin{aligned}\Delta\theta_t &= -\frac{1}{2}\eta \cdot \nabla_\theta(L(\theta_t)) \\ &= \eta \cdot \mathbb{E}_\pi\left[r + \gamma \cdot max_{a'}Q(s', a'; \theta_{t-1}) - Q(s, a; \theta_t)\right] \cdot \nabla_\theta Q(s, a; \theta_t)\end{aligned} \tag{16}$$

Instead of calculating the full expectation in the above gradient, the loss function of the ImprovedDQL-TTA is computationally optimized by stochastic gradient descent [17]. The weights of the ImprovedDQL-TTA approximation are trained using a gradient descent rule, and the parameter θ can be updated using stochastic gradient descent by

$$\begin{aligned}\Delta\theta_t &= \eta \cdot \left(r + \gamma \cdot max_{a'}Q(s', a'; \theta_{t-1}) - Q(s, a; \theta_t)\right) \cdot \nabla_\theta Q(s, a; \theta_t) \\ \theta_t &= \theta_t - \eta \cdot \Delta\theta_t\end{aligned} \tag{17}$$

Stochastic gradient descent is simple and appealing for DQL-TTA; however, it is not sample efficient. In this paper, mini-batch stochastic gradient descent learning is therefore proposed to discover the optimal fitting value function of ImprovedDQL-TTA by training on mini-batch crowdsourcing data. Instead of making decisions based solely on the current allocation experience, the allocation experience replay helps the ImprovedDQL-TTA network to learn from several mini-batches of crowdsourcing data. Each of these allocation experiences is stored as a four-dimensional vector of $\langle state, action, reward, nextstate\rangle$. During training iteration t, allocation experience $e_t = (s_t, a_t, r_t, s_{t+1})$ is stored into a replay tuple $D = \{e_1, ..., e_t\}$. The memory buffer of the allocation experience replay is fixed, and as new allocation experience are inserted, previous experience are removed [19]. To train the ImprovedDQL-TTA neural networks, uniform mini-batches of experiences are extracted randomly from the allocation memory buffer.

To obtain stable Q-values, a separate target network is used to estimate the loss function after every training iterations; another neural network, whose weights are changed gradually compared to the primary Q-network, is also used [35]. In this context, the ImprovedDQL-TTA algorithm learns to optimize two separate neural networks $Q(s, a; \theta)$ and $Q(s, a; \hat{\theta})$ with current learning parameters θ and previous learning parameters $\hat{\theta}$. θ are updated numerous times during the training iterations and are cloned to the previous parameters $\hat{\theta}$ after $NUM_{training}$ iterations.

$$\theta_t = \theta_t - \eta_t \cdot \frac{1}{b}\sum_{t=k}^{k+b}\Delta\theta_t \tag{18}$$

ImprovedDQL-TTA is refreshed with a batch of collected samples in the experience replay buffer by means of mini-batch stochastic gradient descent at each decision epoch.

Theorem 1 (The convergence analysis of mini-batch stochastic gradient descent) *Assume that there are two constants A and B that satisfy* $E[\|\nabla h_b(\theta)\|^2] \leq A$ *and* $\mathbb{E}[\|\theta^* - \theta_t\|^2] \leq B$, *where t denotes the gradient optimization iteration and*

$$\nabla h_b(\theta) = \frac{1}{b}\sum_{t=k}^{k+b} \Delta\theta_t \tag{19}$$

Let $h_{min}(\theta) = min\{h(\theta_1), h(\theta_2), \cdots, h(\theta_t)\}$ and assume that

$$1 > \eta_t > 0, \sum_{t=0}^{\infty} \eta_t^2 < \infty, \sum_{t=0}^{\infty} \eta_t = \infty \tag{20}$$

When the optimization of the mini-batch approach reaches $t+1$ iterations, then

$$\begin{aligned} \|\theta_{t+1} - \theta^*\|^2 &= \|\theta_t - \eta_t \cdot \nabla h_b(\theta) - \theta^*\|^2 \\ &= \|\theta_t - \theta^*\|^2 - 2\eta_t \cdot \nabla h_b(\theta) \cdot (\theta_t - \theta^*) + \eta_t^2 \cdot \|\nabla h_b(\theta)\|^2 \end{aligned} \tag{21}$$

According to the conditional expectation of mathematics, we can obtain

$$\begin{aligned} \mathbb{E}[\|\theta_{t+1} - \theta^*\|^2|\theta_t] &= \mathbb{E}[\|\theta_t - \theta^*\|^2|\theta_t] - 2\eta_t \cdot \mathbb{E}[\nabla h_b(\theta) \cdot (\theta_t - \theta^*)|\theta_t] + \\ &\quad \eta_t^2 \cdot \mathbb{E}[\|\nabla h_b(\theta)\|^2|\theta_t] \\ &\le \|\theta_t - \theta^*\|^2 - 2\eta_t \cdot (h(\theta_t) - h(\theta^*)) + \eta_t^2 \cdot A^2 \end{aligned} \tag{22}$$

Taking the expectation of θ_t in Equation (22) yields

$$\mathbb{E}[\|\theta_{t+1} - \theta^*\|^2] = \mathbb{E}[\|\theta_t - \theta^*\|^2] - 2\eta_t \cdot \mathbb{E}[h(\theta_t) - h(\theta^*)] + \eta_t^2 \cdot A^2 \tag{23}$$

Accordingly,

$$\mathbb{E}[\|\theta_{t+1} - \theta^*\|^2] \le \mathbb{E}[\|\theta_t - \theta^*\|^2] - 2\sum_t \eta_t \cdot \mathbb{E}[h(\theta_t) - h(\theta^*)] + A^2 \cdot \sum_t \eta_t^2 \tag{24}$$

Since $\mathbb{E}[\|\theta_{t+1} - \theta^*\|^2] \ge 0$, we obtain

$$\begin{aligned} \mathbb{E}[\|\theta_{t+1} - \theta^*\|^2] + A^2 \sum_t \eta_t^2 &\ge 2\sum_t \eta_t \cdot \mathbb{E}[h(\theta_t) - h(\theta^*)] \\ &\ge 2\sum_t \eta_t \cdot \mathbb{E}[h_{min}(\theta_t) - h(\theta^*)] \end{aligned} \tag{25}$$

Since $\mathbb{E}[\|\theta_t - \theta^*\|^2] \le B$, we obtain

$$B + A^2 \sum_t \eta_t^2 \ge 2\sum_t \eta_t \cdot \mathbb{E}[h_{min}(\theta_t) - h(\theta^*)] \tag{26}$$

and

$$\mathbb{E}[h_{min}(\theta_t) - h(\theta^*)] \le \frac{B + A^2 \sum_t \eta_t^2}{2\sum_t \eta_t} \tag{27}$$

Since $\sum_{t=0}^{\infty} \eta_t = \infty$, it is clear that $h_{min}(\theta) \to h(\theta^*)$.

Therefore, it can be concluded that ImprovedDQL-TTA with mini-batch stochastic gradient descent converges to $h(\theta^*)$.

ϵ-decreasing greedy policy The ImprovedDQL-TTA algorithm selects the allocation action a with the maximum Q value by *exploiting* the knowledge found by the current

s. To build a better estimate of the optimal ImprovedDQL-TTA function, the algorithm should *explore* and select a different allocation action from the current best allocation. In this paper, the ϵ-greedy policy is employed to select a random allocation action ϵ at one time ($0 \leq \epsilon \leq 1$) and to select the optimal allocation action by maximizing its Q value at the other time [15]. By means of this strategy, ImprovedDQL-TTA can achieve a trade off between exploration and exploitation in uncertain mobile crowdsourcing systems. The ϵ-greedy policy can be illustrated as follows

$$\pi(a|s) = \begin{cases} \frac{\epsilon}{actnum} + 1 - \epsilon, & if\ a^* = argmax_{a \in A} Q(s,a) \\ \frac{\epsilon}{actnum}, & otherwise \end{cases} \tag{28}$$

where *actnum* denotes the total number of available allocation actions.

Theorem 2 (ϵ-greedy policy improvement) *For any ϵ-greedy policy π, the ϵ -greedy policy π' with respect to q_π is an improvement, $v_{\pi'}(s) \geq v_\pi(s)$.*

$$\begin{aligned} q_\pi(s, \pi'(s)) &= \sum_{a \in A} \pi'(a|s) q_\pi(s,a) \\ &= \frac{\epsilon}{actnum} \sum_{a \in A} q_\pi(s,a) + (1-\epsilon) max_{a \in A} q_\pi(s,a) \\ &\geq \frac{\epsilon}{actnum} \sum_{a \in A} q_\pi(s,a) + (1-\epsilon) \sum_{a \in A} \frac{\pi(a|s) - \epsilon/actnum}{1-\epsilon} q_\pi(s,a) \\ &= \sum_{a \in A} \pi(a|s) q_\pi(s,a) \\ &= v_\pi(s) \end{aligned} \tag{29}$$

Therefore, the ϵ-greedy policy is an improvement, $v_{\pi'}(s) \geq v_\pi(s)$.

To maintain a good balance of exploration and exploitation, a suitable learning parameter should be selected for the ϵ-greedy strategy. In the early training time, a more random policy should be used to encourage initial exploration, and as training time progresses, a more greedy policy should be considered. The training performance of ImprovedDQL-TTA can be improved by using an ϵ-greedy parameter that changes during training, which is defined as following.

$$\epsilon = \epsilon - \frac{\epsilon_i - \epsilon_f}{explore} \tag{30}$$

where ϵ_i is the initial value of ϵ, ϵ_f is the final value of ϵ, and *explore* is the total number of training steps.

Novel neural network architecture with action advantage function To further improve the convergence stability, a novel deep network architecture is integrated into ImprovedDQL-TTA for learning the crowdsourcing decision process with an action advantage function [33–35]. The key idea of this mechanism is to design a novel neural network with two sequences of fully connected layers. In this way, the state values and the action advantage are separately learned by the novel ImprovedDQL-TTA neural network. Figure 4 illustrates the novel neural network architecture.

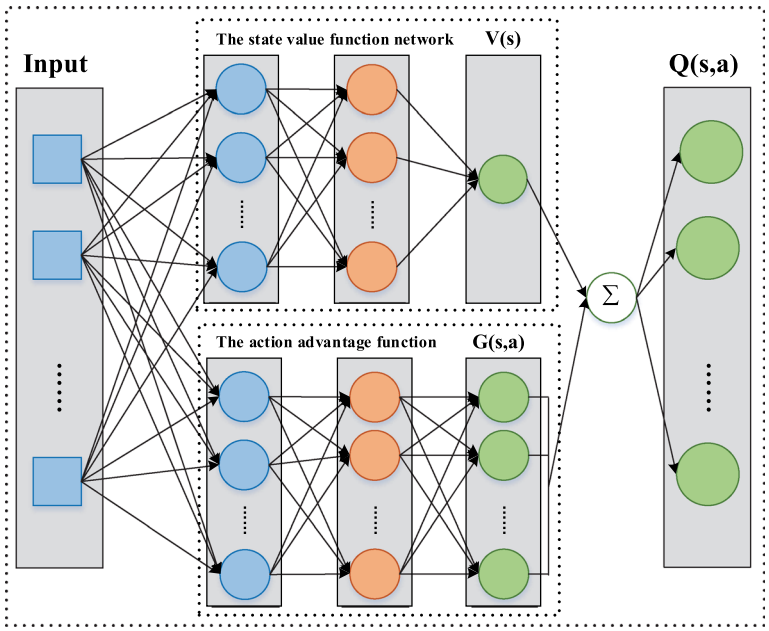

Fig. 4 The novel neural network architecture of ImprovedDQL-TTA

For a stochastic policy π, $Q_\pi(s,a)$ and $V_\pi(s)$ can be formulated as

$$\begin{aligned} Q_\pi(s,a) &= \mathbb{E}[R_t|s_t = s, a_t = a, \pi] \\ V_\pi(s) &= \mathbb{E}_{a\sim\pi(s)}[Q^\pi(s,a)] \end{aligned} \tag{31}$$

The action advantage function can be defined as

$$G_\pi(s,a) = Q_\pi(s,a) - V_\pi(s) \tag{32}$$

Note that $\mathbb{E}[G_\pi(s,a)] = 0$. Intuitively, the $V_\pi(s)$ function calculates the value of a particular state s, and $Q_\pi(s,a)$ evaluates the value of selection action a in state s and then combines the results to estimate the crowdsourcing action value. Based on this definition, the evaluation of the relative importance of the each crowdsourcing action can be obtained from the action advantage function $G_\pi(s,a)$.

To estimate the values of V and G functions, ImprovedDQL-TTA is implemented with a novel neural network, where two streams of fully connected layers output vector $V(a;\beta)$ and vector $G(s,a;\alpha)$. ImprovedDQL-TTA combines $V_\pi(s)$ and $G_\pi(s,a)$ to obtain $Q_\pi(s,a)$, as follows

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + G(s,a;\theta,\alpha) \tag{33}$$

and

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + \left(G(s,a;\theta,\alpha) - \max G(s,a;\theta,\alpha)\right) \tag{34}$$

where α and β are parameters of the two sequences of novel neural network layers. The action advantage function has zero advantage in selecting an action. For $a^* = argmax_{a\in A}Q(s,a;\alpha,\beta) = argmax_{a\in A}G(s,a;\alpha)$, the function obtains

$Q(s, a^*; \alpha, \beta) = V(s; \beta)$. Furthermore, for better stability, an alternative module of ImprovedDQL-TTA replaces the max operator with an average operator

$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + \left(G(s, a; \theta, \alpha) - \frac{1}{|A|} \sum_{a} G(s, a; \theta, \alpha) \right) \tag{35}$$

ImprovedDQL-TTA is an intelligent algorithm for addressing sequential decision-making problems of mobile crowdsourcing systems. ImprovedDQL-TTA is implemented with mini-batch stochastic gradient descent, ϵ-decreasing greedy policy, and a novel network architecture with an action advantage function. To intelligently develop an appropriate strategy, ImprovedDQL-TTA is built with a multiple-layer network that takes the crowdsourcing state encoded in a $[1 \times statenum]$ vector and learns the best action (mobile workers), mapping all possible actions in a vector of length *actnum*. In summary, the pseudo code for improved deep Q-learning-based trust-aware task allocation is illustrated in Algorithm 1.

Algorithm 1 ImprovedDQL-TTA

Load the mobile crowdsourcing environment: $env = crowdsourcing.make()$
$snum = env.observationspace.n$
$actnum = env.actionspace.n$
Initialize crowdsourcing replay memory D
Build the deep neural network for Q-learning-based trust-aware task allocation:
  $dqn = DQN(\eta, \gamma, statenum, actnum, \epsilon)$
Initialize primary and target neural networks with random weights
for each episode in range(episodes) **do**
  Get the initial state: $s = env.reset()$
  while TRUE **do**
    Initialize the crowdsourcing environment: $env.render()$
    Select action $action$ by means of random ϵ-policy derived from
      $action = dqn.epsilon\_greedy(s)$
      $\epsilon = \epsilon - \frac{\epsilon_i - \epsilon_f}{explore}$
    Execute action $action$, compute reward:
      $s\_, reward, done = env.step()$
    Store each experience $(state, action, reward, s\_)$ in its experience buffer:
      $dqn.store\_experience(s, action, r_t, s\_)$
    Sample batch memory from all memory:
      $indices = random.choice(batchsize)$
      $batch = memory[indices, :]$
    Calculate Q value by combining the value function and action advantage function
      $Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + \left(G(s, a; \theta, \alpha) - \frac{1}{|A|} \sum_a G(s, a; \theta, \alpha) \right)$
    Update Q value y_t as:

$$y_t = \begin{cases} r_t, & if A(s\_) = \varnothing \\ r_t + \gamma \cdot max_{a_{t+1}} Q(s\_, a_{t+1}; \theta_{t-1}, \alpha_{t-1}, \beta_{t-1}), & else \end{cases}$$

    Perform a gradient descent step on: $loss = \left(y_t - Q(s, a; \theta_t, \alpha_t, \beta_t)\right)^2$
    Perform function $GradientDescentOptimizer(loss, batch)$
    $S \leftarrow s\_$
    if done == True **then**
      break
    end if
  end while
end for

Table 2 Parameter settings for the synthetic data set

Synthetic data set	Gowalla data set
Latitude $\sim \mu(37.71, 37.83)$, *longitude* $\sim \mu(-122.37, -122.50)$	38,333 task locations
1000 mobile workers	1083 users
Latitude $\sim \mu(37.71, 37.83)$, *longitude* $\sim \mu(-122.37, -122.50)$	227,428 worker locations

ImprovedDQL-TTA is able to effectively identify an optimal solution for the large-scale MCMDP. ImprovedDQL-TTA operates by learning to optimize the expected reward of selecting an action for a given state and discovering the optimal action-selection policy to stably adapt to changes in a large-scale environment.

Experimental results and analysis

The prototype applications were programmed on the JetBrains PyCharm Community platform. All the algorithms were implemented in Python 3.5 programming language, and the experiments were run on 64 bit windows 10 with an Intel(R) Core(TM) i5-7300HQ@2, 50 GHz, 16 GB of RAM, and 500 GB disk storage. The performance of the proposed ImprovedDQL-TTA algorithm was compared to the reference algorithms. A series of experiments were performed on synthetic data from the real world. In this section, computer simulations are conducted to illustrate the performance of the proposed ImprovedDQL-TTA algorithm in mobile crowdsourcing systems. We first present the experimental setting; then, the performance under different scenarios simulated and analyzed. Finally, the convergence of the proposed ImprovedDQL-TTA algorithm is illustrated.

Experimental setting

Existing research has addressed spatial task allocation by simulating mobile crowdsourcing environments by means of experimental data sets. Data sets from location-based social networks have been used to evaluate dynamic crowdsourcing algorithms. A similar approach is followed here to evaluate the performance of the proposed algorithm. The experimental data set is presented and evaluated in the following subsections.

The synthetic data set consists of real-world data obtained from Gowalla, a popular location-based social network. Gowalla was selected as our experimental data set for evaluating ImprovedDQL-TTA, and San Francisco was chosen as the experimental region, within the boundary [37.709, 37.839, −122.373, −122.503]. The Gowalla data set includes check-ins by a large number of users at numerous locations in San Francisco. The data set comprises 1,083 persons, 38,333 locations, and 227,428 check-ins. For synthetic experimentation purposes, the task and worker locations were randomly initialized with *latitude* $\sim \mu(37.71, 37.83)$ and *longitude* $\sim \mu(-122.37, -122.50)$. Table 2 summarizes both data sets used for the data-driven initialization [6].

Figure 5 illustrates the geographical map and its data table for mobile crowdsourcing systems in San Francisco.

The synthetic data were used to study the proposed algorithm. Users in the Gowalla data set are regarded as mobile workers, and the locations and check-ins are initialized

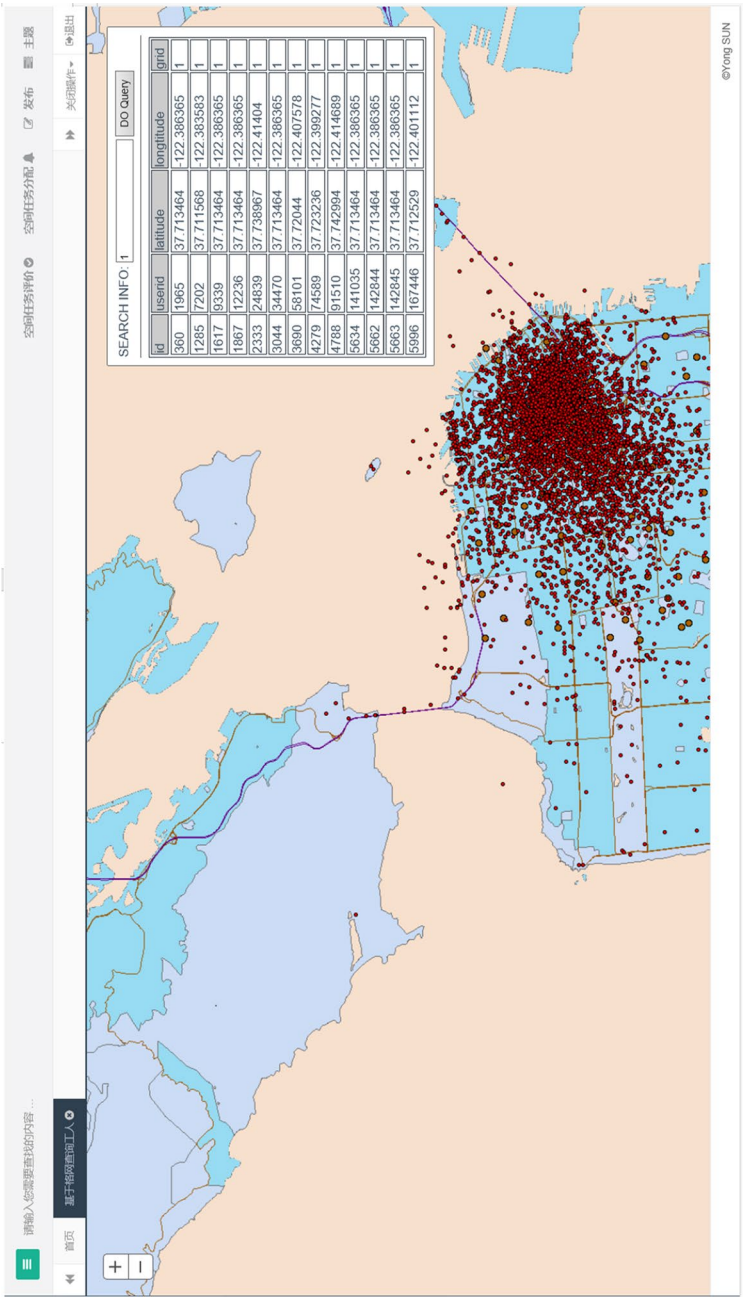

Fig. 5 Mobile crowdsourcing System data in San Francisco

Table 3 Mobile crowdsourcing settings for scalability evaluation

Setting content	Setting values
Number of spatial tasks	10, 50, 100
Number of mobile workers	10, 50, 100

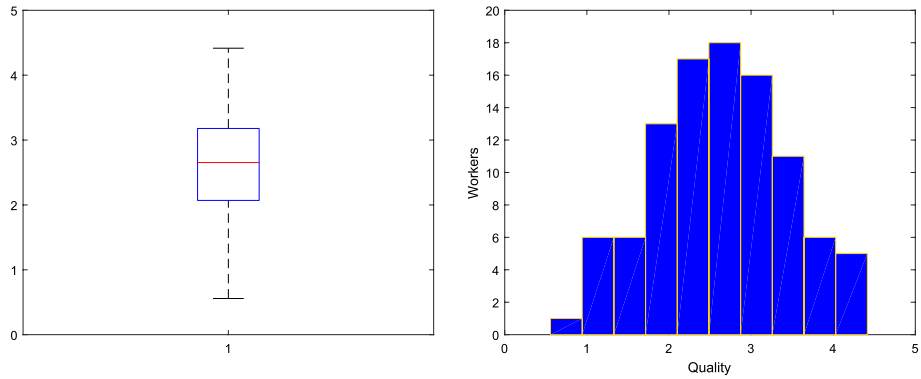

Fig. 6 Distribution of mobile worker quality data

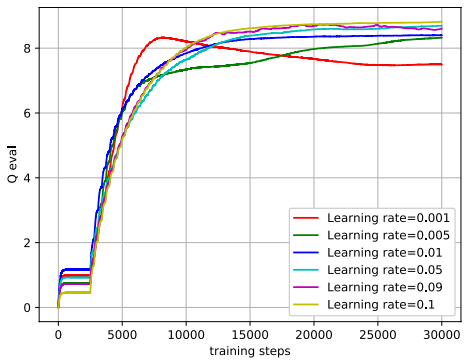

Fig. 7 Efficiency of ImprovedDQL-TTA with different learning rates

in relation to the mobile users. Mobile crowdsourcing requesters are randomly generated by sampling Gowalla check-ins [6]. To evaluate the scalability of our proposed ImprovedDQL-TTA, the mobile crowdsourcing parameters are set as in Table 3.

Furthermore, the trustworthiness of mobile workers is evaluated in terms of their trust value tr_j, sampled from the parameterized uniform distribution, that is, $tr_j \sim \mu(tr_{min}, tr_{max})$. For workers, the qualities of the tasks are also randomly generated from a uniform distribution, that is, $sw_{i,j} \sim \mu(tr_j, 0.1)$. We set the mobile worker trust parameters as tr, and the trust score range is [0.5, 1], [1, 2], [2, 3], [3, 4], [4, 4.5). To satisfy the experimental requirements, we evaluate our proposed ImprovedDQL-TTA algorithm on synthetic and real-world data sets. The quality data for each mobile worker are simulated in the MCMDP model with a random vector. The parameters of the quality vector are obtained from a Gaussian distribution. Figure 6 illustrates the quality distribution of mobile workers.

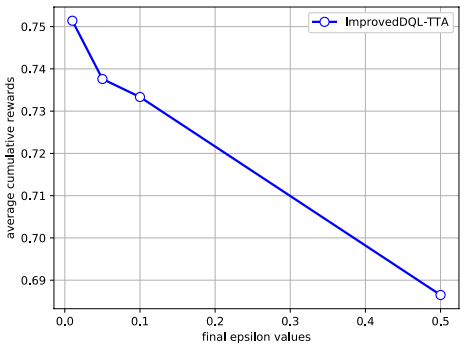

Fig. 8 Efficiency of ImprovedDQL-TTA with different greedy rate

Table 4 Learning parameter settings

Setting content	Setting parameter	Setting value
Learning rate	η	0.1
Initial epsilon value	ϵ_i	0.9
Final epsilon value	ϵ_f	0.01

With the purpose of solving dynamic decision problems in uncertain mobile crowdsourcing, the ImprovedDQL-TTA algorithm iteratively runs until convergence. As the trust score and travel distance of mobile workers are dynamic, the trust and distance values of 10% of the mobile workers are regenerated periodically every 30,000 episodes.

Algorithm parameter study

The parameters of the algorithm are defined for our experiments to ensure high-quality crowdsourcing solutions, and the number of iterations is set to 30,000. This section discusses two core parameters of ImprovedDQL-TTA: the learning rate α and ϵ-greedy rate. The following experiments investigate the two learning parameters.

Experiment 1: ImprovedDQL-TTA learning rate evaluation. To improve the learning efficiency of the proposed algorithm, a suitable learning rate must be seleted. This experiment varies the learning rate η to investigate the learning efficiency. As shown in Fig. 7, when $\eta = 0.001$, the Q value continues its downward trend after approximately 20000 iterations, which indicates that the learning with this parameter setting is inefficient; when $\eta = 0.005$, the Q value are continue their upward trend after approximately 30,000 iterations, which indicates $\eta = 0.005$ results in inefficient learning; when $\eta = 0.09$, the Q values reach to a maximum after approximately 15,000 iterations, but the learning value is not stable; when alpha=0.1, the Q values rapidly reach to the maximum after around 15,000 iterations, and the final Q values with $\eta = 0.1$ is higher than the that of the other η settings.

Experiment 2: ImprovedDQL-TTA ϵ-greedy rate evaluation. In this experiment, the number of spatial tasks is set to 50, the number of candidate mobile workers is set to 50, and the greedy rate ϵ is varied. To investigate the impact of the greedy rate on the proposed ImprovedDQL-TTA algorithm, the final epsilon parameter of ϵ-greedy

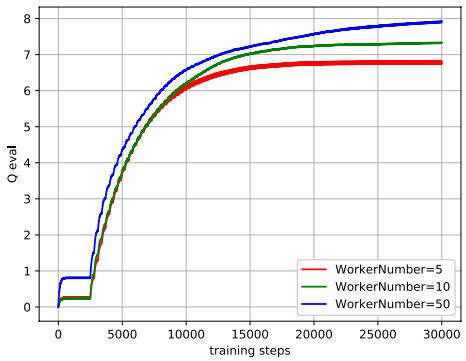

Fig. 9 Efficiency of ImprovedDQL-TTA with different worker number

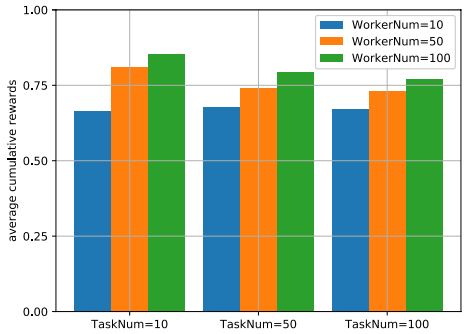

Fig. 10 The average rewards of ImprovedDQL-TTA with different worker and task scale

is varied in [0.01, 0.05, 0.1, 0.5]. Figure 8 illustrates the average cumulative rewards with different final ϵ values. From the results, we can make an observation that the learning quality with $\epsilon_f = 0.01$ is higher than other ϵ_f settings.

In the ImprovedDQL-TTA algorithm, the learning rate η is defined as 0.1, the initial ϵ -greedy value is defined as 0.9, and the final ϵ-greedy value is defined as 0.01 as shown in Table 4.

ImprovedDQL-TTA performance study

Experiment 3: Learning efficiency with different numbers of worker. As illustrated in Fig. 9, the Q value performance is evaluated with respect to the number of mobile workers. The number of mobile workers is varied in [5, 10, 50], and the spatial task number is set to 50.

Figure 9 shows that the Q values increase with increasing number of mobile workers because a greater number of mobile workers increases the chances of selecting better workers.

Experiment 4: Average rewards with different worker and task scales. As illustrated in Fig. 10, the average cumulative reward performance is estimated with respect to the number of mobile workers. The number of mobile workers varies in [10, 50, 100], and the number of spatial tasks is varied in [10, 50, 100].

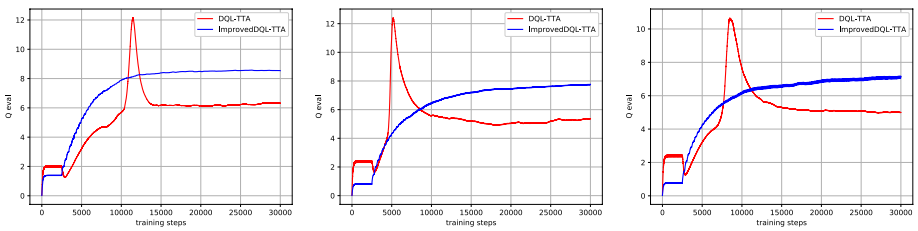

Fig. 11 Learning Q value evaluation of ImprovedDQL-TTA and DQL-TTA

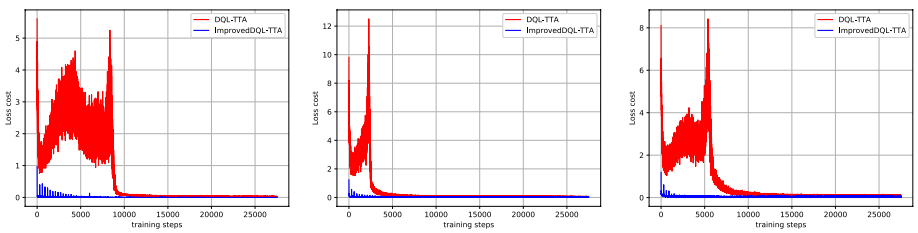

Fig. 12 Learning loss evaluation of ImprovedDQL-TTA and DQL-TTA

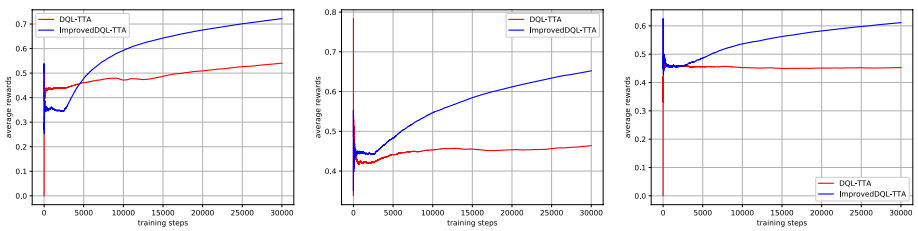

Fig. 13 Learning the average rewards of ImprovedDQL-TTA and DQL-TTA

Figure 10 illustrates that the average cumulative rewards increase with increasing number of candidate mobile workers because a greater number of workers increases the chances of selecting higher-quality workers by using ImprovedDQL-TTA.

Experiment 5: Comparison of DQL-TTA and ImprovedDQL-TTA. The mobile crowdsourcing scale is denoted by the number of spatial tasks. The experimental number of mobile workers is set to 50, and the number of required spatial tasks are set to [10, 50, 100]. The experiment creates 10×50, 50×50, and 100×50 task-and-worker pairs matrix for comparing the proposed algorithm ImprovedDQL-TTA and DQL-TTA. The x-axes indicates the training steps.

Evaluation of Q values: The proposed ImprovedDQL-TTA algorithm is run in this setting and the Q values are compared with those of DQL-TTA. Figure 11 shows the Q value results for the proposed ImprovedDQL-TTA algorithm with DQL-TTA algorithm. The ImprovedDQL-TTA algorithm consistently produces higher Q values than the DQL-TTA algorithm after approximately 15,000 iterations.

Evaluation of training loss cost: The proposed ImprovedDQL-TTA algorithm is run in this setting and the training lost costs are compared with those of DQL-TTA. Figure 12 illustrates the loss function cost results of the proposed ImprovedDQL-TTA algorithm and DQL-TTA algorithm. The two algorithms converge within the certain number of

Fig. 14 Learning performance with different worker

training steps. Therefore, the learning accuracies of ImprovedDQL-TTA gradually improves as training progresses.

Evaluation of the average accumulated reward: The proposed ImprovedDQL-TTA algorithm is run in the setting of experiment 5 and the average accumulated reward is compared with that of DQL-TTA. Figure 13 illustrates the average accumulated award results of the proposed ImprovedDQL-TTA algorithm and DQL-TTA algorithm. The ImprovedDQL-TTA algorithm consistently leads to higher rewards than the DQL-TTA algorithm during the training, which indicates the proposed ImprovedDQL-TTA algorithm produces a better allocating solution in uncertain mobile crowdsourcing systems.

Efficiency evaluation of ImprovedDQL-TTA: We compare our proposed ImprovedDQL-TTA algorithm to DQL-TTA in terms of run-time performance. According to the experimental requirements, the number of spatial tasks is set to 50, and the number of available mobile workers is varied in [10, 50, 100]. The proposed ImprovedDQL-TTA algorithm is run in this setting. Figure 14 illustrates the run-time performance of the proposed ImprovedDQL-TTA algorithm in comparison to that of DQL-TTA when varying the number of mobile worker. The blue bar describes the average run-time of the ImprovedDQL-TTA with different mobile worker scales. As illustrated in the figure, ImprovedDQL-TTA is more efficient than the DQL-TTA algorithm in terms of calculation time.

The above experiments illustrate the performance of the proposed ImprovedDQL-TTA in terms of Q value, loss cost, average accumulated reward and run time. The experimental results on the data sets of uncertain mobile crowdsourcing illustrated that ImprovedDQL-TTA algorithm outperformed DQL-TTA algorithm. Therefore, our proposed ImprovedDQL-TTA produces better solutions than DQL-TTA. Moreover, ImprovedDQL-TTA is much more stable when solving large-scale MCMDP problems of uncertain mobile crowdsourcing systems. Given enough iterations, the ImprovedDQL-TTA algorithm will converge to the optimal Q value. Therefore, ImprovedDQL-TTA can learn to optimize its efforts to solve the dynamic trust-aware task allocation problems in an adaptive and effective manner.

Conclusion

Due to the advancing technology of smart phones with numerous built-in sensors, mobile crowdsourcing has recently promoted the combination of collective intelligence beyond geographical boundaries. Mobile workers need to collaborate with other

workers for accomplishing multiple tasks. Trustworthiness is considered as a key factor in mobile crowdsourcing to enable effective collaboration. In this paper, a new and highly effective learning algorithm has been proposed to process dynamic Trust aware Task Allocations (TTA) in uncertain mobile crowdsourcing systems. Specifically, the TTA optimization problem, which aims at maximizing trust score and minimizing the travel distance cost, is formulated as Mobile Crowdsourcing Markov Decision Process (MCMDP). Furthermore, to solve the large-scale MCMDP problem, an Improved Deep Q-Learning-based Trust aware Task Allocation (ImprovedDQL-TTA) algorithm is proposed as an improvement over trust collaboration optimization modelling in uncertain crowdsourcing systems. The proposed algorithm combines both trust aware task allocation optimization and deep Q-Learning techniques. The theoretical analysis was conducted to prove the applicability of ImprovedDQL-TTA. Experimental simulations were carried out to establish the obvious advantage of our proposed algorithm through comparisons with the reference algorithm. The ImprovedDQL-TTA algorithm exhibits distinct advantages that make it effective to large-scale spatial collaboration problems in uncertain mobile crowdsourcing systems.

Acknowledgements
The authors thank the reviewers for their suggestions which helped in improving the quality of the paper.

Authors' contributions
Yong Sun contributed to the original idea, algorithm and the whole manuscript writing. Wenan Tan supervised the work and helped with revising and editing the manuscript. Both authors read and approved the final manuscript.

Funding
This work was supported in part by the National Natural Science Foundation of China under Grant No. 61272036, Anhui Provincial Natural Science Foundation under Grant No. 1908085MF191, and the University Natural Science Foundation of Jiangsu Province under Grant No. 18KJB520008.

Availability of data and materials
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1] Anhui Center for Collaborative Innovation in Geographical Information Integration and Application, Chuzhou University, Chuzhou 239000, Anhui, China. [2] Anhui Engineering Laboratory of Geoinformation Smart Sensing and Services, Chuzhou University, Chuzhou 239000, Anhui, China. [3] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, Jiangsu, China.

Received: 23 January 2019 Accepted: 22 June 2019
Published online: 04 July 2019

References

1. Ju R, Zhang Y, Zhang K (2015) Exploiting mobile crowdsourcing for pervasive cloud services: challenges and solutions. IEEE Commun Mag 53(3):98–105
2. Hassan UU, Curry E (2016) Efficient task assignment for spatial crowdsourcing: a combinatorial fractional optimization approach with semi-bandit learning. Expert Syst Appl 58((C)):36–56
3. To H (2016) Task assignment in spatial crowdsourcing: challenges and approaches. In: Proceedings of the 3rd ACM SIGSPATIAL PhD symposium, San Francisco, CA, USA, 26 June–1 July 2016
4. Tran L, To H, Fan L (2018) A real-time framework for task assignment in hyperlocal spatial crowdsourcing. ACM Trans Intell Syst Technol 9(3):37:1–37:26
5. Li Y, Shin B (2017) Task-management method using R-tree spatial cloaking for large-scale crowdsourcing. Symmetry 9(12):311
6. Sun Y, Wang J, Tan W. Dynamic worker-and-task assignment on uncertain spatial crowdsourcing. In: IEEE CSCWD 20th international conference on computer supported cooperative work in design, 2018. pp 755–760
7. Guo B, Liu Y, Wang L (2018) Task allocation in spatial crowdsourcing: current state and future directions. IEEE Intern Things J 5(3):1749–1764
8. Wang Y, Cai Z, Tong X (2018) Truthful incentive mechanism with location privacy-preserving for mobile crowdsourcing systems. Comput Netw 135:32–43

9. Zhao Y, Han Q (2016) Spatial crowdsourcing: current state and future directions. IEEE Commun Mag 54(7):102–107
10. Liu A, Wang W, Shang S (2018) Efficient task assignment in spatial crowdsourcing with worker and task privacy protection. GeoInformatica 22(2):335–362
11. Chi Z, Wang Y, Huang Y (2018) The novel location privacy-preserving CKD for mobile crowdsourcing systems. IEEE Access 6:5678–5687
12. Kazemi L, Shahabi C, Chen L, et al. GeoTruCrowd:trustworthy query answering with spatial crowdsourcing. In: ACM Sigspatial international conference on advances in geographic information systems. ACM, 2013. pp 314–323
13. Hayam M, Sonia BM, Omar H (2015) Trust management and reputation systems in mobile participatory sensing applications: a survey. Comput Netw 90:49–73
14. Watkins CJ (1989) Learning from delayed rewards. Ph.D. thesis. Cambridge University, Cambridge
15. Sutton RS, Barto AG (1988) Reinforcement learning: an introduction, vol 1. MIT Press, Cambridge
16. Azevedo CR, Von Zuben FJ (2015) Learning to anticipate flexible choices in multiple criteria decision-making under uncertainty. IEEE Trans Cybern 46(3):778–791
17. Mnih V et al (2015) Human-level control through deep reinforcement learning. Nature 518:529–533
18. Silver D et al (2017) Mastering the game of go without human knowledge. Nature 550(7676):354–359
19. Liu N, et al (2017) A hierarchical framework of cloud resource allocation and power management using deep reinforcement learning. In: Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS), Atlanta, GA, USA, pp 372–382
20. Sun Y, Peng M, Mao S (2018) Deep reinforcement learning based mode selection and resource management for green fog radio access networks. IEEE Intern Things J 99:1
21. Chittilappilly AI, Chen L, Ameryahia S (2016) A survey of general-purpose crowdsourcing techniques. IEEE Trans Knowl Data Eng 28(9):2246–2266
22. Doan A, Ramakrishnan R, Halevy AY (2011) Crowdsourcing systems on the World-Wide Web. ACM, New York
23. Whitehill J, Wu TF, Bergsma J, Movellan JR, Ruvolo PL (2009) Whose vote should count more: optimal integration of labels from labelers of unknown expertise. In: Advances in neural information processing systems. pp 2035–2043
24. Parameswaran A, Sarma AD, Garcia-Molina H (2011) Human-assisted graph search: it's okay to ask questions. Proc Vldb Endow 4(5):267–278
25. Liu Xuan, Meiyu Lu, Ooi Beng Chin, Shen Yanyan, Sai Wu, Zhang Meihui (2012) CDAS: a crowdsourcing data analytics system. Proc VLDB Endow 5(10):1040–1051
26. Bulut MF, Yilmaz YS, Demirbas M. Crowdsourcing location-based queries. In: 2011 IEEE international conference on pervasive computing and communications workshops (PERCOM Workshops). IEEE, New York, pp 513–518
27. Sun Y, Tan W, Li LX (2016) A new method to identify collaborative partners in social service provider networks. Inform Syst Front 18(3):565–578
28. Awal GK, Bharadwaj KK (2014) Team formation in social networks based on collective intelligence-an evolutionary approach. Appl Intell 41(2):627–648
29. Miao C, Yu H, Shen Z (2016) Balancing quality and budget considerations in mobile crowdsourcing. Decis Support Syst 90:56–64
30. Feng Z, Zhu Y, Zhang Q et al (2014) Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing. In: Proceedings of the IEEE INFOCOM conference, pp 1231–1239
31. Kazemi L, Shahabi C (2012) GeoCrowd: enabling query answering with spatial crowdsourcing. In: Advances in geographic information systems. pp 189–198
32. Cheng P, Lian X, Chen L et al (2017) Prediction-based task assignment in spatial crowdsourcing. In: International conference on data engineering, pp 997–1008
33. Schaul T, Quan J, Antonoglou I, et al (2016) Prioritized experience replay. In: International conference on learning representations, ICLR
34. Wang Z, Schaul T, Hessel M, et al (2016) Dueling network architectures for deep reinforcement learning. In: International conference on machine learning, pp. 1995–2003
35. Li Y (2017) Deep reinforcement learning: an overview
36. Tan W, Sun Y, Li L (2014) A trust service-oriented scheduling model for workflow applications in cloud computing. IEEE Syst J 8(3):868–878

Publisher's Note