Context-Aware Reliable Crowdsourcing in Social Networks

Jiuchuan Jiang¹⁰, Bo An, Yichuan Jiang¹⁰, Senior Member, IEEE, and Donghui Lin

Abstract—There are two problems in the traditional crowdsourcing systems for handling complex tasks. First, decomposing complex tasks into a set of micro-subtasks requires the decomposition capability of the requesters; thus, some requesters may abandon using crowdsourcing to accomplish a large number of complex tasks since they cannot bear such heavy burden by themselves. Second, tasks are often assigned redundantly to multiple workers to achieve reliable results, but reliability may not be ensured when there are many malicious workers in the crowd. Currently, it is observed that the workers are often connected through social networks, a feature that can significantly facilitate task allocation and task execution in crowdsourcing. Therefore, this paper investigates crowdsourcing in social networks and presents a novel context-aware reliable crowdsourcing approach. In our presented approach, the two problems in traditional crowdsourcing are addressed as follows: 1) the complex tasks can be performed through autonomous coordination between the assigned worker and his contextual workers in the social network; thus, the requesters can be exempt from a heavy computing load for decomposing complex tasks into subtasks and combing the partial results of subtasks, thereby enabling more requesters to accomplish a large number of complex tasks through crowdsourcing, and 2) the reliability of a worker is determined not only by the reputation of the worker himself but also by the reputations of the contextual workers in the social network; thus, the unreliability of transient or malicious workers can be effectively addressed. The presented approach addresses two types of social networks including simplex and multiplex networks. Based on theoretical analyses and experiments on a real-world dataset, we find that the presented approach can achieve significantly higher task allocation and execution efficiency than the previous benchmark task allocation approaches; moreover, the presented contextual reputation mechanism can achieve relatively higher reliability when there are many malicious workers in the crowd.

Index Terms—Context-aware, crowdsourcing, reliability, social networks, task allocation, task execution.

Manuscript received May 30, 2017; revised August 31, 2017; accepted November 17, 2017. Date of publication December 19, 2017; date of current version January 15, 2020. The work of Y. Jiang was supported in part by the National Natural Science Foundation of China under Grant 61472079 and Grant 61170164, and in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20171363. This paper was recommended by Associate Editor W. Hsu. (*Corresponding author: Jiuchuan Jiang.*)

J. Jiang and B. An are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: jiangjiuchuan@163.com; boan@ntu.edu.sg).

Y. Jiang is with the School of Computer Science and Engineering, Southeast University, Nanjing 211189, China (e-mail: yjiang@seu.edu.cn).

D. Lin is with the Department of Social Informatics, Kyoto University, Kyoto 606-8501, Japan (e-mail: lindh@i.kyoto-u.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2017.2777447

I. INTRODUCTION

CROWDSOURCING is a task allocation paradigm in which a requester allocates tasks to a group of workers chosen from a population [1], [2]. There are many crowdsourcing platforms oriented to microtasks, such as Amazon's Mechanical Turk, MicroWorkERs, and ShortTask [3]. Generally, the microtasks-oriented crowdsourcing platforms have two general limitations: 1) the complex tasks cannot be executed directly and 2) the workers may be transient and unreliable. Existing studies often addressed the two limitations as follows: 1) complex tasks are decomposed to a set of micro-subtasks that can be solved in multiple phases [2] and 2) reliability is achieved by assigning each task redundantly to multiple workers [4], [5]. However, the existing studies have the following problems.

- 1) In the existing studies of crowdsourcing complex tasks in which the tasks are decomposed to a set of interdependent micro-subtasks, the requesters must determine the optimal task decomposition. Thus, some requesters may abandon using crowdsourcing to accomplish a large number of complex tasks since they cannot bear such heavy burden by themselves. Moreover, the offline decomposition of tasks may not match the real-time situation of workers because the available workers may be dynamically changed at some crowdsourcing platforms [6].
- 2) In the existing studies of achieving reliability by redundantly assigning each task to multiple workers, the reliability may not be ensured when there are many malicious workers in the crowd [7], [8]. Although some studies introduced the reputation mechanism to cope with malicious workers [9], reputation may be undeterminable in traditional crowdsourcing platforms due to the transient characteristics of the workers.

Nowadays, with the significant development of social networks [16], [32], the crowd of workers is often connected by the social networks [11]–[13], [40], [41] and many related studies have harnessed the crowdsourcing power of social media [14], [15]. For example, social networks such as Facebook and Twitter can also be considered as crowd providers [34]. Specifically, the crowdsourcing power of social networks has been used for disaster relief [14]. Consider the following motivating scenario.

In the Lushan earthquake that occurred on April 20, 2013, in China, many donated supplies such as tents and food were aggregated in nearby cities, but it was difficult to deliver

2168-2216 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

these supplies to the rural villages. Therefore, the charitable organization called upon volunteers to help deliver supplies at some Chinese social media platforms such as Weibo and Renren. Usually, more than one person is required to deliver a block of tents; the required group includes one driver with a truck, a local guide, and several porters. Thus, if a volunteer wished to undertake a delivery task, he would find other partners in social networks and they would make the delivery together. For example, say there is a truck driver who offers, through a social media platform, to deliver and who is approved by the charitable organization, he will then seek other trusted partners, including a local guide and several porters, through his social network. After the truck driver finds the necessary partners, they can cooperate to complete the delivery task. Finally, the truck driver and his partners achieve certain reputations through completing the task, and they can apply for new delivery tasks by virtue of their reputations.

Generally, the advantages of social networks for crowdsourcing include at least the following.

- 1) The workers within a social network are more inclined to cooperate with each other to execute a complex task [32].
- 2) It is easier to find professional and helpful workers through social networks because real-world social networks include many professional groups [31] and many users of social networks undertake outsourced tasks not only for monetary return but also due to interest or willingness to share with their friends [13].
- 3) The social networks can broadcast outsourced tasks more rapidly, thereby hastening the completion of tasks [14], [16].

In fact, there are now some related studies that are often implemented by team formation in which a team of workers that can perform outsourced tasks is found through social networks [17], [18]. However, the requesters must undertake heavy computing loads for selecting appropriate team members [17]. Moreover, these related studies often assume that the team members are reliable, which may sometimes not conform to the actual situation [18].

To address the two critical problems of traditional crowdsourcing systems that are not solved by existing studies on crowdsourcing in social networks, this paper presents an approach for context-aware reliable crowdsourcing in social networks. The context of a worker in the social network can be defined as the counterparts that interact with this worker through the social network [19]. In our approach, when a requester wishes to outsource a task, a worker candidate's self-situation and his contextual-situation in the social network are considered.

With our approach, the two problems in the existing studies are addressed in the social network environment as follows.

 The requester only needs to assign tasks directly to the workers without decomposing the tasks; the assigned workers will then start to execute the complex tasks by autonomously coordinating with other workers in the social network context. Therefore, the requester avoids heavy computing load when decomposing complex tasks, thereby enabling more requesters to accomplish a large number of complex tasks through crowdsourcing. Moreover, our approach is implemented by the autonomous coordination among workers, which offers an appropriate fit to real situations in which the set of available workers may be dynamically changed.

2) The reliability of a worker is determined not only by the reputation of the worker himself but also by the reputations of his contextual workers. Even when a worker comes to the system for the first time, his initial reputation can be measured through the past experiences of his contextual workers. Therefore, reliability of the transient workers can be achieved.

Our context-aware crowdsourcing approach involves assigning a principal worker who will then recruit other assistant workers from the social network, which should satisfy the following general objectives: the communication cost and the reservation wages are minimized and the total reputations are maximized. This optimization problem is proved to be NPhard in Section III-C; thus, a heuristic approach that can be realized in reasonable time complexity is then presented. In the approach, a concept of crowdsourcing value is defined to measure the probability of a worker being assigned a task when the context of the worker in the social network is considered. In this way, workers with higher crowdsourcing values can be preferentially selected to perform the task. The experimental results show that the presented approach can achieve higher task allocation and execution efficiency than the previous benchmark approaches and can achieve higher reliability by adopting the contextual reputation mechanism.

The remainder of this paper is organized as follows. In Section II, we compare this paper with the related work on the subject. In Section III, we present the problem description. In Section IV, we present the context-aware task allocation model. In Section V, we present the context-aware task execution model. In Section VI, we present the reward mechanism. In Section VII, we provide the experimental results. Finally, in Section VIII, we conclude this paper.

II. RELATED WORK

A. Crowdsourcing for Complex Tasks

Traditional crowdsourcing platforms, such as Amazon's Mechanical Turk, are often oriented to microtask markets. A popular method of performing complex tasks is to decompose the task into a flow of simple subtasks and then combine the partial results of the subtasks to obtain the final answer [2]. Tran-Thanh *et al.* [2] proposed the first crowdsourcing algorithm, BudgetFix, to solve the complex tasks that involve various types of interdependent microtasks structured into complex workflows. Moreover, Dai *et al.* [21] used Bayesian network learning and partially observable Markov decision processes to make dynamic control for workflow optimization.

Team formation is another method that can be used to crowdsource complex tasks. In this method, individuals with different skills form a team that completes the tasks. In many of the existing studies, the team formation is controlled by the requester, and interested candidate workers advertise their skills and bid prices for their participation into the team. Liu *et al.* [18] presented an efficient method for team formation in crowdsourcing which are implemented through some profitable and truthful pricing mechanisms. Kargar *et al.* [22] presented a method for finding a team of experts that covers all the skills for the task and minimizes communication cost.

Overall, the studies described above may place heavy computing loads on requesters, rendering them inappropriate when the number of complex tasks is large [3].

B. Reliable Crowdsourcing

Workers may be transient and unreliable [7]. Therefore, it is necessary to ensure reliability in crowdsourcing to make the workers really work on the task [8].

A typical solution is redundantly assigning each task to more than one worker and combing the answers by measures such as majority voting [4]. Karger *et al.* [4] presented an algorithm for deciding which tasks to assign to which workers and for inferring correct answers from the workers' answers. However, this type of approach may be infeasible when many malicious workers exist.

Another typical solution for reliable crowdsourcing is using a trust and reputation mechanism. Ren *et al.* [40] integrated the social relationship and reputation management into mobile crowdsourcing and proposed a social aware crowdsourcing with a reputation management scheme, which can efficiently improve the crowdsourcing utility and the quality of sensing reports. Venanzi *et al.* [5] addressed the problem of fusing untrustworthy answers provided by a crowd of workers and incorporated the trust model into a fusion method. Zhang and van der Schaar [9] proposed protocols to incentivize workers to perform tasks well and reliably by using a novel game-theoretic model and integrating reputation mechanisms.

Unlike these studies, this paper integrates the redundancy mechanism and the reputation mechanism. Moreover, the reliability of a worker in this paper will be determined not only by the reputation of the worker himself but also by the reputations of the contextual workers. Even when a worker comes to the system for the first time, his initial reputation can be measured through the past experiences of his contextual workers. This can solve the problem that the reputation may be undeterminable due to the transient characteristics of the workers [9], [23].

C. Crowdsourcing in Social Networks

The existing related studies can be categorized into two classes. The first class of studies mainly considers how to harness the crowdsourcing power of social media [14], [15], and how to use social networks as crowdsourcing platforms [34], and the second class mainly considers how to find a group of workers in social networks to complete outsourced tasks [11], [17].

In the first class of studies, Gao *et al.* [14] addressed harnessing the crowdsourcing power of social media for disaster relief. Ren *et al.* [41] exploited the capabilities of mobile device users connected via wireless networks to form a mobile cloud to provide pervasive crowdsourcing services.

Lim *et al.* [15] combined the power of crowdsourcing with that of social networks and provided a Web-based tool for the automation of stakeholder analysis. Here, the social networks were considered as new crowdsourcing platforms, but the allocation and reliability of complex tasks were seldom investigated systematically.

In the second class of studies, Chamberlain [11] proposed a definition for groupsourcing that includes the idea that a group of people connected through a social network is used to complete a task. However, in this class of studies, the requesters must undertake heavy computing loads to coordinate the organization of workers.

D. Context-Aware Crowdsourcing and Other Task Allocation

There are some studies on context-aware crowdsourcing. Tamilin et al. [23] presented a prototype implementation of a context-aware mobile crowdsourcing system that makes it possible to conduct crowdsourcing campaigns with users carrying mobile devices. Rana et al. [37] developed a contextaware crowdsourcing method by using smart phones for noise mapping, which can provide a feasible platform to assess noise pollution. Hu et al. [38] presented a multidimensional social network architecture for mobile crowdsensing, which enables context awareness in the mobile crowdsensing applications. Alt et al. [39] combined the Web-based crowdsourcing and user-generated content to integrate location as a context parameter for distributing tasks to workers. In summary, the existing studies of context-aware crowdsourcing mainly focused on the context of mobile devices, but they did not address the context of social networks among workers and the reliability in context-aware crowdsourcing.

Task allocation and self-organization in open environments have been investigated in the previous studies [35], [36]. Jiang *et al.* [24] explored the context-aware task allocation in social networks. The problems in [24] differ from those addressed in this paper. This paper aims to maximize the probability of task completion through autonomous cooperation of workers in social networks. This is an important and realistic feature of crowdsourcing, as stated in Section I. However, the studies in [24] focus only on minimizing resource access time in social networks, a feature that is not crucial in crowdsourcing.

III. PROBLEM DESCRIPTION

To clearly illustrate the research problem in this paper, we will first introduce the original framework for reliable allocation of simple tasks in crowdsourcing systems in Section III-A. Then, in Section III-B, we introduce the motivation and state of our research problem. Finally, we analyze the complexity of the problem in Section III-C. Table I summarizes the notations used in this paper.

A. Original Framework for Reliable Allocation of Simple Tasks in Crowdsourcing Systems

A simple task can be completed by one worker independently. Given a budget b_t for a simple task t, the necessary skills to complete t are represented by S_t . Let there be a crowd

Notation	Definition	Notation	Definition
t	Task t	d _{ij}	The communication distance between workers w_i and w_j in the social network
b _t	The budget for task <i>t</i>	$ au_{wi}$	The threshold of w_i to decide whether to accept the request from another worker for assistance in executing a task
S_t	The set of skills required by task <i>t</i>	N_k	Network layer <i>k</i>
W	A crowd of workers	$N(w_i)$	The network layers involving worker w_i
W_t	The set of workers assigned for task <i>t</i>	d_{ij^k}	The communication distance between w_i and w_j in the network layer N_k .
w_i	Worker i	τ_{wi}^k	Worker w_i 's threshold for k-type immediate social links
Ywi	The reservation wage of w_i	$CNv_k(t)$	The contextual crowdsourcing value of N_k for task t
S_{wi}	The set of skills of w_i	ni←j	The number of w_i 's real assistance for w_i 's executing tasks
R_{wi}	The reputation of w_i	$c_j(\leftarrow i)$	The credit of w_j paid by w_i
Com _{wi}	The communication cost between w_i and his contextual workers	$C_{i \rightarrow j}(t)$	The credit paid by w_i to w_j for executing task t
CS_{wi}	The set of skills of the contextual workers of w_i	$m_{i \rightarrow j}(t)$	The monetary reward paid by w_i to w_j
CR_{wi}	The contextual reputation of w_i	$v_j(i-t)$	The assistance value of w_j perceived by w_i for executing task t
$v_i(t)$	The self-owned crowdsourcing value of worker w_i for task t	τ_{wj}^{k}	Worker w_j 's threshold for the requests from k -type links
$Cv_i(t)$	The contextual crowdsourcing value of worker w_i for task t	$\tau_j(i)$	The threshold of worker w_i for worker w_i 's request

TABLE I DEFINITIONS OF NOTATIONS

of workers, W. Then, the simple task allocation is to redundantly assign the task to multiple workers under the budget constraint to improve the accuracy of the result [2]. This can be defined as follows: the requester or system assigns t to a set of workers, W_t , $W_t \subseteq W$; $\forall w_i \in W_t$, w_i can satisfy the skills required for task t and complete t independently; and the sum of the reservation wages of the workers in W_t does not exceed b_t .

Moreover, the reputation mechanism can be used to encourage workers to complete the assigned tasks reliably [9]. The reputation of a worker is mainly determined by the worker's past experiences in completing tasks; if a worker has richer experience of successful completion of tasks, his reputation is higher, and vice versa.

Therefore, the objective of reliable task allocation in general crowdsourcing systems is to select a set of workers that maximizes the following values.

- The degree of redundancy, which denotes that the system redundantly assigns task *t* to as many workers as possible (each of whom fully possesses the skills required for *t*) under the constraint of the budget.
- 2) The reputations of the assigned workers.

Given a simple task t; $\forall w_{i \in} W$, the set of skills of w_i is S_{wi} , and the reservation wage of w_i is γ_{wi} . The objective of reliably allocating t can then be formalized as selecting a set of workers W_t that can satisfy the following:

$$W_{t} = \underset{W_{t} \subseteq W}{\operatorname{arg\,max}} \left(\alpha_{1} \cdot \overbrace{\left(\frac{|W_{t}|}{\sum_{\forall w_{i} \in W_{t}} \gamma_{w_{i}}}\right)}^{\operatorname{Redundancy}} + \alpha_{2} \cdot \left(\overbrace{\sum_{\forall w_{i} \in W_{t}} R_{w_{i}}}^{\operatorname{Reputations}}\right) \right)$$
(1)

subject to
$$\forall w_i \in W_t \land S_{w_i} \supseteq S_t$$
 (2)

$$\sum_{V_{w_i} \in W_t} \gamma_{w_i} \le b_t \tag{3}$$

where α_1 and α_2 are used to determine the relative importance of the two factors.

B. Motivation and Problem Statement of Complex Task Crowdsourcing in Social Networks

We will now present some notable observations made in related studies to motivate this paper on crowdsourcing in social networks. Gray et al. [13] observed four popular crowdsourcing platforms namely, MTurk, UHRS, LeadGenius, and Amara, and found that the crowd of workers is actually a rich collaborative network and that workers often communicate via phone, forums, chat, Facebook, or in person, to share information about tasks and requesters. Yin et al. [12] specifically observed the collaboration network of workers on the MTurk platform, in which they executed a task where over 10 000 workers self-reported their communication links to the other workers. They found that there is a substantial communication network within the crowd of workers that is related to the workers' usage on the online forum. Therefore, these observations motivate this paper on crowdsourcing in social networks.

In current crowdsourcing markets, there are many complex tasks. A complex task involves many computational operations and may not be completed independently by a nonprofessional individual worker. Therefore, the situation in which each assigned worker can fully cover the required skills of *t*, i.e., $\forall w_i \in W_t \land S_{wi} \supseteq S_t$ in (2) cannot be satisfied. Because the self-owned skills of the assigned worker may only partially cover the necessary skills required by a complex task, the situation should be revised to " $\forall w_i \in W_t \land S_{wi} \cap S_t \neq \phi$."

The new issues of complex task crowdsourcing in social networks can be described as follows.

 An assigned worker may not perform the complex task alone and should coordinate with other workers in the social network to request assistance in the lacking skills. Therefore, when the crowdsourcing system wishes to assign a task t to a worker w_i , it considers not only the individual skills of w_i but also the skills of the contextual workers of w_i , CS_{wi} (considering contextual skills in task allocation).

- 2) Because an assigned worker's performance in executing a task is influenced by his coordination with the contextual workers, a worker's reliability is determined not only by his own reputation but also by the reputations of his contextual workers. Therefore, the reputation R_{wi} in (1) should consider this factor (*considering contextual reputations in task allocation*).
- 3) Because the assigned workers can coordinate autonomously with the contextual workers in the social network to execute complex tasks, the system or the requester does not need to decompose the complex task into microtasks. However, an efficient coordination mechanism between the assigned worker and the contextual workers should be designed (coordinating with contextual workers in task execution).
- 4) To promote the cooperation of workers in executing tasks, a proper reward mechanism should be designed that encourages not only the workers to accept the tasks by themselves but also to contribute skills to assist others in executing tasks. In addition, the reward mechanism considers not only the monetary reward but also the reputation reward (*rewarding contextual workers after task execution*).

Moreover, communication costs among workers may significantly influence the performance in social networks [22]. Therefore, we should consider the communication costs of each assigned worker, w_i , with his contextual workers (which can be denoted by Com_{wi}).

We now extend the original framework in Section III-A by considering these ew issues. Then, the objective of reliable task allocation of crowdsourcing in social networks is to select a set of workers that maximizes the following values: 1) the coverage degree of each assigned worker's contextual skills for the task's required skills; 2) the contextual reputation of each assigned worker; 3) the degree of redundancy; and 4) the inverse of the communication cost of each assigned worker with his contextual workers. Therefore, the objective can now be formalized as selecting a set of workers W_t that can satisfy the following:

$$W_{t} = \underset{W_{t} \subseteq W}{\operatorname{arg\,max}} \left(\underbrace{\left(\underbrace{\sum_{\forall w_{i} \in W_{t}} \left| CS_{w_{i}} \bigcap S_{t} \right| \right)}_{\forall w_{i} \in W} + \underbrace{\sum_{\forall w_{i} \in W_{t}}^{\operatorname{Contextual reputations}} \left(\underbrace{\left(\underbrace{\sum_{\forall w_{i} \in W_{t}} \left| CS_{w_{i}} \bigcap S_{t} \right| \right)}_{\forall w_{i} \in W} + \underbrace{\sum_{\forall w_{i} \in W_{t}}^{\operatorname{Redundancy}} \left(\underbrace{\sum_{\forall w_{i} \in W_{t}} \left| V_{w_{i}} \right|}_{\forall w_{i} \in W_{t}} \right) / \underbrace{\sum_{\forall w_{i} \in W_{t}}^{\operatorname{Conmunication costs}}}_{\forall w_{i} \in W_{t}} \right)} \right)$$
(4)

subject to: $(\forall w_i \in W_t) \land \left(S_{w_i} \bigcap S_t \neq \phi\right)$ (5)

$$\sum_{\forall w_i \in W_t} \gamma_{w_i} \le b_t.$$
(6)

With the above objective function, the requester can assign the task to the workers to maximize the possibility that the task's required skills can be satisfied and the accuracy of execution results. Moreover, the communication cost for executing the task will be minimized, which is also focused in the previous benchmark studies on task allocation in the social networks [26].

Then, each assigned worker, $w_i(\forall w_i \in W_t)$, will coordinate with his contexts to execute *t*. Finally, the system will reward w_i and his contextual workers for executing *t* after *t* is completed.

Our approach includes the following three components.

- Context-aware task allocation, which aims to assign the task to workers who can maximize the combination of contextual skills, contextual reputations, and redundancy and minimize the communication costs, as shown in Section IV.
- Context-aware task execution, which describes how each assigned worker coordinates autonomously with his contextual workers in the social network to execute complex tasks, as shown in Section V.
- Reward after task execution, which describes how to distribute rewards among the assigned workers and their contextual workers, as shown in Section VI.

C. Complexity Analyses

As stated above, the core of our research problem mainly involves assigning a principal worker who will then recruit other assistant workers from the social network. Thus, the set of workers participating in the task includes the assigned principal worker and the assistant workers, which should possess all of the skills required by the task and satisfy the following conditions: 1) the communication costs among participating workers are minimized; 2) the total reservation wages of the participating workers are minimized; and 3) the total reputations of the participating workers are maximized.

Theorem 1: Let there be a crowd of workers W that is organized in a social network. Assigning a set of participating workers including a principal worker and some assistant workers from W to cover the skills required by a task t and satisfy the above three conditions is NP-hard.

Proof Sketch: Our research problem includes three independent subproblems that can be described to assign a set of participating workers within a social network to achieve the following three objectives: 1) minimizing the total communication costs; 2) minimizing the total reservation wages; and 3) maximizing the total reputations. The first subproblem has already been proven in the previous benchmark studies to be NP-hard [22], [25], [26]. Since our research problem involves this NP-hard subproblem in combination with two other independent subproblems, we have Theorem 1.

Because the research problem is NP-hard, we can present a heuristic approach that can be realized in reasonable time complexity but not present an approach to compute the optimal solution. In our heuristic approach, we define a concept of crowdsourcing value that combines the factors in (4) to measure the probability of a given worker's being selected to participate in a task. Moreover, the principal worker will find assistant workers within the social network using the breadthfirst search method, which can effectively reduce the costs of communication between the principal worker and the assistant workers.

IV. CONTEXT-AWARE TASK ALLOCATION

This section investigates how to allocate a task to workers by considering their contextual situations within their social networks. We first design a metric of contextual crowdsourcing value to measure the probability of a given worker's being assigned the task. We then present algorithms for allocating the task to workers to maximize the contextual crowdsourcing values, which address the simplex and multiplex social networks, respectively.

A. Contextual Crowdsourcing Value

Each worker has three properties: 1) skills; 2) reputation; and 3) reservation wage. Among these three factors, the skills determine the worker's capacity to complete a task, the reputation determines the worker's reliability in completing the task, and the reservation wage is the minimum wage a worker is willing to accept as compensation in exchange for completing a crowdsourcing task [10], [20].

The probability of a given worker being assigned a task can be defined as the crowdsourcing value of the worker. To optimize the first three factors in (4), this value is defined to be determined by the following attributes: 1) the coverage degree of the worker's skills for the necessary skills required by the task; 2) the reputation of the worker; and 3) the occupancy rate of the worker's wage in the task's budget.

Definition 1 (Self-Owned Crowdsourcing Value of a Worker): Given a budget b_t for a task t, the set of necessary skills to complete t is S_t . The self-owned crowdsourcing value of a worker, w_i , for a task t is defined as

$$v_i(t) = \frac{\alpha_1(|S_{w_i} \cap S_t|/|S_t|) + \alpha_2(R_{w_i})}{\alpha_3(\gamma_{w_i}/b_t)}$$
(7)

where S_{wi} , R_{wi} , and γ_{wi} denote the skills, reputation, and reservation wage of w_i , respectively, and α_1 , α_2 , and α_3 are the parameters that determine the relative importance of the three factors.

In the social network context, when a worker receives a request from another worker for assistance in executing a task, the worker can decide whether to accept the request according to the benefits associated with the task and the worker's own expectation threshold. Therefore, each worker w_i has a predefined threshold τ_{wi} ; the worker will accept a request to assist another worker only if the benefit associated with the ongoing task exceeds the threshold. The threshold of a worker is related not only to the worker's reservation wage but also to other factors such as reputation and credit; this will be shown in Section V-A.

To optimize the four factors in (4), the contextual crowdsourcing value of w_i can be defined to be determined by the crowdsourcing value of w_i himself and the following attributes of the contextual workers: 1) the coverage degree of the contextual workers' skills for current lacking skills of w_i for t; 2) the reputations and thresholds of the contextual workers; and 3) the communication distance between w_i and the contextual workers.

Definition 2 (Contextual Crowdsourcing Value of a Worker in a Simplex Social Network): The contextual crowdsourcing value of a worker w_i for a task t in a simplex social network is defined as

$$Cv_{i}(t) = \beta_{1} \cdot v_{i}(t) + \beta_{2}$$

$$\times \sum_{\forall w_{i} \in (W - \{w_{i}\})} \left(\frac{\alpha_{1}(|(S_{w_{j}} - S_{i}) \cap S_{t}|/|S_{t}|) + \alpha_{2}(R_{w_{j}})}{\alpha_{3}(\tau_{w_{j}})} / d_{ij} \right)$$
(8)

where W denotes the crowd of workers in the social network and d_{ij} denotes the communication distance between w_i and w_j in the social network. The communication distance can be defined as the length of the shortest path between the two workers in the network. β_1 and β_2 are used to determine the relative importance of context.

Moreover, in reality, the workers may be associated with multiplex social networks [27]. Each worker has different biases for accepting requests from different types of social networks. For example, a scientist may accept requests for cooperation from other workers in a scientific collaboration network more readily than he or she may accept requests from workers within a tour social network. Therefore, each worker has more than one threshold, and each threshold is specific for a type of social link. Let the social links among workers be classified into λ different types $1, \ldots, \lambda$; then, each worker w_i has a threshold τ_{wi}^k for *k*-type immediate social links, $k = 1, \ldots, \lambda$.

Definition 3 (Network Layer in a Multiplex Network [27]): Let the multiplex network be $N = \langle W, E \rangle$, where W denotes the crowd of workers and E denotes the set of social links among the workers. N can then be divided into λ network layers. Each network layer, N_k , $1 \leq k \leq \lambda$, is defined as follows:

$$N = \left\{ N_k | N_k = \langle W_k, E_k \rangle \land W_k \subseteq W \land E_k \subseteq E \\ \land \left(\forall e_i^k, e_j^k \in E_k \Rightarrow l_{e_{ik}} = l_{e_{jk}} \right) \right\}$$
(9)

where e_i^k and e_j^k are the links in the network layer N_k , l_{eik} and l_{ejk} denote the link types of e_i^k and e_j^k . The network layers involving worker w_i are denoted as

The network layers involving worker w_i are denoted as $N(w_i)$

$$N(w_i) = \left\{ N_k \middle| N_k \in N \land N_k = \langle W_k, E_k \rangle \land w_i \in W_k \right\}.$$
(10)

The contextual crowdsourcing value of a worker in a multiplex social network is determined by the sum of the ones in all network layers. Therefore, we have the following definition.

Definition 4 (Contextual Crowdsourcing Value of a Worker in a Multiplex Social Network): The contextual crowdsourcing value of a worker w_i for a task t in a multiplex social network is defined as

$$Cv_{i}(t) = \beta_{1} \cdot v_{i}(t) + \beta_{2} \cdot \sum_{\forall N_{k} \in N(w_{i})} \sum_{\forall w_{j} \in (W_{k} - \{w_{i}\})} \left(\frac{\alpha_{1}(|(S_{w_{j}} - S_{w_{i}}) \cap S_{t}| / |S_{t}|) + \alpha_{2}(R_{w_{j}})}{\alpha_{3}(\tau_{w_{j}}^{k})} / d_{ij}^{k} \right) (11)$$

where d_{ij}^k denotes the communication distance between w_i and w_j in the network layer N_k .

Lemma 1: Let there be two workers, w_i and w_j . If $Cv_i(t) > Cv_j(t)$, it is more probable that task *t* will be assigned to w_i by a crowdsourcing system in social networks to achieve the task allocation objective in (4).

Proof Sketch: The three factors that influence the task allocation objective in (4) are determined as follows: the coverage degree for the skills required by a task is determined by $|S_{wi} \cap S_t|/|S_t|$ and $|(S_{wj} - S_{wi}) \cap S_t|/|S_t|$, the reputations of the workers executing the task are determined by R_{wi} and R_{wj} , and the redundancy of the assigned workers is determined by (γ_{wi}/b_t) . Because $Cv_i(t) > Cv_j(t)$, the comprehensive value of the three factors for w_i is higher than that of w_j ; accordingly, w_i can satisfy the objective in (4) to achieve the target W_t better than w_j . Therefore, it is more probable that task t will be assigned to w_i .

B. Task Allocation Mechanism

In our model, two types of workers participate in the tasks: one type is the *principal worker*, who is assigned by the system to be responsible for a task; the other type is the *assistant workers* who are autonomously sought by the principal worker in the context of the social network. The assignment of the principal worker is implemented by the task allocation model in this section, and the selection of the assistant workers is implemented by the task execution model described in Section V.

1) Task Allocation in Simplex Social Networks: In the allocation of task *t*, we will select the candidate worker from the crowd who has the highest contextual crowdsourcing value for *t* and whose reservation wage does not exceed *t*'s budget; the task will then be assigned to this candidate worker. We can repeat the allocation process to redundantly assign the task to other workers until the budget of the task has been used up.

To avoid the situation in which some workers are heavily loaded for a task, we make the following assumption in simplex social networks: each worker can perform only one role in a task, i.e., if a worker is assigned as the principal worker for a task, he or she cannot serve as an assistant worker (requested by another assigned principal worker) for the task. Therefore, when the system calculates a worker's contextual crowdsourcing value for task t, the workers already assigned to t will be excluded from the context.

The task allocation can be described as Algorithm 1. The time complexity of Algorithm 1 is $O(|W|^2)$, where |W| is the number of workers in the social networked crowd.

Theorem 2: Let the set of assigned workers for task t using Algorithm 1 be W_t . It is, then, assumed that there is another worker set in the crowd, W'_t , and that the total reservation

Algorithm 1: Task Allocation Based on Workers' Contextual Crowdsourcing Values (*t*, *W*)

1 $b_1 = 0; W_t = \{\}; W_{temp1} = W;$ 2 While $(b_1 == 0)$ do: $\forall w_i \in W_{temp1}$: 3 $Cv_i(t) = \beta_1 \cdot v_i(t) + \beta_2$ $\cdot \sum_{\forall w_i \in (W_{temp1} - \{w_i\})} \left(\frac{\alpha_1 \left(|(S_{w_j} - S_{w_i}) \bigcap S_t| / |S_t| \right) \cdot \alpha_2(R_{w_j})}{\alpha_3(\tau_{w_j})} / d_{ij} \right);$ $w_* = \arg \max_{\forall w_i \in W_{temp1}} (Cv_i(t)); \ b_2 = 0; \ W_{temp2} = W_{temp1};$ 4 If $\gamma_{W*} > b_t$: $b_2 = 1$; 5 While $((W_{temp2} \neq \{\}) \text{ and } (b_2 == 1))$ do: 6 $\mathbf{W}_{temp2} \stackrel{!}{=} \mathbf{W}_{temp2} - \{w_*\};$ 7 If $W_{temp2} \neq \{\}$: 8 $w_* = \arg \max_{\forall w_i \in W_{temp2}} (Cv_i(t));$ 9 If $\gamma_{w*} \le b_t$: $b_2 = 0$; 10 If $W_{temp2} \neq \{\}$: /*Allocate a worker*/ 11 $W_t = W_t \bigcup \{w_*\}; \ b_t = b_t - \gamma_{w*};$ 12 $W_{temp1} = W_{temp1} - \{w_*\};$ 13 **Else**: $b_1 = 1$; 14 If $b_t \leq 0$: $b_1 = 1$; /*the budget is used up*/ 15 **16 Output** (*W_t*); /**The final set of assigned workers**/ 17 End.

wages of all workers in W'_t do not exceed b_t . Thus, we have

$$\begin{pmatrix} \forall W'_t \land W'_t \subseteq W \land \sum_{\forall w_i \in W'_t} \gamma_{w_i} \leq b_t \end{pmatrix}$$

$$\Rightarrow \left(\frac{\sum_{\forall w_i \in W_t} CV_i(t)}{|W_t|} \geq \frac{\sum_{\forall w_j \in W'_t} CV_j(t)}{|W'_t|} \right).$$
(12)

Proof: We can use *reductio ad absurdum* to prove Theorem 1. Assume that there is a set of workers $W'_t, W'_t \neq W_t$, that is assigned by the system to task *t* and that the total reservation wages of all workers in W'_t do not exceed b_t . If the assumption

$$\left(\sum_{\forall w_i \in W_t} CV_i(t)\right) / |W_t| < \left(\sum_{\forall w_j \in W_t'} CV_j(t)\right) / |W_t'|$$

is true, then there exists at least one worker with a higher contextual crowdsourcing value and whose reservation wage does not exceed the remaining budget of t but which cannot be selected by Algorithm 1, and another worker with lower contextual crowdsourcing value will be assigned to t. However, from steps 4 and 10, in each round for selecting the assigned worker, the worker with the highest contextual crowdsourcing value whose reservation wage does not exceed the remaining budget of t will be the first to be definitely assigned to t. Therefore, the above assumption cannot occur in reality when Algorithm 1 is used.

Theorem 2 ensures that the set of assigned workers with the highest average contextual crowdsourcing values can be achieved by Algorithm 1. Then, according to Lemma 1, the objective of task allocation of crowdsourcing in social networks can be approached most efficiently.

2) Task Allocation in Multiplex Social Networks: For the task allocation in multiplex social networks, there are two

methods: one is the method based on workers' contextual crowdsourcing values, which can be implemented using Algorithm 1 by revising the calculation of $Cv_i(t)$ in step 3 (i.e., $Cv_i(t)$ is calculated according to Definition 4); another method is based on both network layers' and workers' contextual crowdsourcing values. Now we introduce the second method in detail.

The network layer-based task allocation method was initially presented in [27]; in this method, the proper network layers are allocated to a task, and the real assigned agents are then selected within the allocated network layers. We now extend the method to the allocation of crowdsourcing tasks in multiplex social networks. To measure the probability of a given network layer's being assigned a task, we define the contextual value of the network layer as the sum of ones of all workers within the layer.

Definition 5 (Contextual Crowdsourcing Value of a Layer in a Multiplex Network): Let the multiplex social network, $N = \langle W, E \rangle$, be classified into λ network layers. For each network layer N_k , $N_k \in N \land N_k = \langle W_k, E_k \rangle$ ($1 \le k \le \lambda$), the contextual crowdsourcing value of N_k for task t is defined as

$$CNv_k(t) = \sum_{\forall w_i \in W_k} Cv_i(t).$$
(13)

Let N_* be the network layer with the highest value, let the task be assigned to the worker with the highest contextual crowdsourcing value in N_* , and let the worker's reservation wage be less than the task's budget; such a process will be repeated until the budget is used up or all the workers in the network layer are considered. To encourage cooperation within a network layer, we make the following assumption, which differs from the assumption in Section IV-B1: each worker can behave as two roles for a task, i.e., if a worker is selected as the assigned worker for a task, he or she can also serve as the assistant worker for another assigned worker within the network layer for the task.

After all the workers in the first selected network layer are considered, the network layer with the second-highest value will be selected if the remaining budget of t is greater than zero. The assigned workers will then be selected from the second network layer. This process will be repeated for other network layers according to their crowdsourcing values in descending order, until the budget of t is used up.

Finally, the task allocation based on both network layers' and workers' contextual crowdsourcing values can be described as Algorithm 2, where W is the crowd of workers in the multiplex social network.

The time complexity of Algorithm 2 is $O(\lambda \cdot |W_k|)$, where λ is the number of network layers and $|W_k|$ is the number of workers in the network layer N_k . The time complexity of Algorithm 1 is $O(|W|^2)$, where |W| is the number of workers in the whole network. In reality, λ is much less than |W| and $|W_k|$ is also less than |W|, thus the time complexity by using Algorithm 2 is less than the one by using Algorithm 1 in multiplex social networks. Therefore, although Algorithm 2 cannot effectively ensure the optimal task allocation result by comparing with Algorithm 1, Algorithm 2 can also have the application value since it can save allocation time significantly.

Algorithm 2: Task Allocation Based on Both Network Layers' and Workers' Contextual Crowdsourcing Values (*t*, *W*)

```
1 \forall w_i \in W: calculate Cv_i(t) by considering w_i's context in W according
   to Equation (11):
 2 \forall N_k \in N: calculate CNv_k(t) according to Equation (13);
3 b_1 = 0; W_t = \{\}; N_{temp} = N;
 4 While (b_1 == 0) do:
         N_* = \arg \max_{\forall N_k \in N_{temp}} (CNv_k(t)); /*N_* = < W_*, E_* > */
5
         b_2 = 0; W_{temp} = W_*;
 6
 7
         While (b_2 = 0) do:
8
               w_* = \arg \max_{\forall w_i \in W_*} (Cv_i(t));
               If \gamma_{W*} \leq b_t:
 9
                     W_t = W_t \bigcup \{w_*\}; b_t = b_t - \gamma_{w*};
10
                     If b_t \le 0: {b_1 = 1; b_2 = 1;}
11
               W_{temp} = W_{temp} - \{w_*\};
12
              If W_{temp} == \{\}: b_2 = 1;
13
14
         N_{temp} = N_{temp} - \{N_*\};
         If N_{temp} == \{\}: b_1 = 1;
15
16 Output (W_t); /*W<sub>t</sub> is the final set of assigned workers*/
```

V. Context-Aware Task Execution

After the task is assigned to a worker through the task allocation approach described in Section IV, the assigned worker will start to execute the task. If the assigned worker cannot execute the task by himself, he will execute the task with the coordination of contextual workers.

A. Preliminaries

17 End.

The workers in a social network are often coadjutant [28], [31]. A worker w_i will have certain obligations to provide assistance for another worker, w_j , if w_j has provided assistances for w_i in the past. Therefore, w_j may accept the requests of w_i for assistance even if the monetary reward provided by w_i is less than w_j 's reservation wage because w_j expects to get the possible assistance from w_i in the future. To measure the obligation to provide assistance between two workers, we define the credit between them to be determined by their cooperation history.

Definition 6 (Credit Between Two Workers): Let there be two workers, w_i and w_j . We use $n_{i \leftarrow j}$ to denote the number of w_j 's real assistance for w_i 's executing tasks. The credit of w_j paid by w_i is in proportion to $n_{i \leftarrow j}$

$$c_{i}(\leftarrow i) = f(n_{i \leftarrow i}) \tag{14}$$

where *f* is a monotonically increasing function. Obviously, the higher $c_j(\leftarrow i)$ is, the more compulsory that w_i should provide assistance for w_j 's request even if w_j cannot provide enough monetary reward to w_i , the reason is that in the past w_j has provided lots of assistance for w_i so that now w_i is obligated to compensate w_j . This credit is different from reputation because the credit involves only two workers whereas the reputation of a worker is perceived by all workers.

If the assigned worker w_i lacks the necessary skills required by task t, i.e., $S_t - S_{wi} \neq \phi$, w_i will seek the assistance of other contextual workers to provide the skills w_i lacks. When w_i requests assistance from another worker, assuming w_j , w_i will offer two items to w_i . 1) The Possible Monetary Reward for Executing Task t: Let $\overline{S_t}$ be the set of skills required for t that are currently lacking and S_{wj} be the set of skills possessed by worker w_j . Therefore, the possible skill contribution of w_j for t is $S_{w_j} \cap \overline{S_t}$. Let the reservation wage of w_i for task t be γ_{wi} . The possible monetary reward paid by w_i to w_j is

$$m_{i \to j}(t) = \lambda \cdot \gamma_{w_i} \cdot \left| S_{w_j} \bigcap \overline{S_t} \right| / |S_t|$$
(15)

where $0 \le \lambda \le 1$, which denotes the percentage of reservation wage that w_i is willing to distribute to other assistant workers.

 The Credit Paid by w_i to w_j for Executing Task t, c_{i→j}(t): If c_{i→j}(t) is high and w_j hopes to obtain assistance from w_i in the future, w_j may accept the current request from w_i even if w_j cannot receive a satisfactory monetary reward for this request. If w_j accepts the request from w_i, then: c_i(← j) = c_i(← j) - c_{i→j}(t), c_j(← i) = c_j(← i) + c_{i→j}(t).

Then, w_j will decide whether to accept the request from w_i for assistance in executing task *t* according to the following four conditions: 1) the possible monetary reward for executing task *t*, $m_{i\rightarrow j}(t)$; 2) the credit paid by w_i to w_j for executing task *t*, $c_{i\rightarrow j}(t)$; 3) the total credits of w_i paid by w_j in the past, $c_i(\leftarrow j)$; and 4) the reputation of w_i , R_i .

In the real world, each person may cooperate initially with his neighbors, and he will then cooperate with other people according to the breadth-first search in the social network [29]. Therefore, we now have the following definition.

Definition 7 (Coordination Tree in the Social Network for a Task): Let w_i be the assigned worker for task t. If w_i requests assistance from other workers in the social network, then the interaction relations between w_i and other workers form a tree whose root is w_i and the depth of each worker in the tree is the shortest interaction distance between w_i and the worker in the worker in the social network. Obviously, the coordination tree can be constructed based on the breadth-first traversal method for the social network without considering the link types, and the workers can be decomposed into varying levels such that the shortest path from w_i to each worker (assuming w_j) in level L_x is with distance x, i.e., $d_{ij} = x$.

Example 1: In Fig. 1, there are three types of links. Let w_6 be the assigned worker. We first compute the varying orders of coordination workers in the social network without considering the link types. Finally, the coordination tree is achieved.

B. Task Execution in Simplex Social Networks

Let the threshold of w_j be τ_{wj} . Worker w_j will accept the request of w_i if the following condition can be satisfied:

$$\left(\eta_1 \cdot m_{i \to j}(t) + \eta_2 \cdot c_{i \to j}(t) + \eta_3 \cdot c_i(\leftarrow j) + \eta_4 \cdot R_{w_i}\right) \ge \tau_{w_j}$$
(16)

where η_1 , η_2 , η_3 , and η_4 are four parameters that are used to determine the relative importance of the four factors.

To optimize the four factors in (4), we can define the assistance value of a worker to be determined by the four attributes in Definition 2 and the credit between the assigned worker and the assistant worker.



Fig. 1. Example coordination tree in a multiplex social network.

Definition 8 (Assistance Value of a Worker in Simplex Network): Let w_i be the assigned worker for task t. If $\overline{S_t}$ is the set of skills for t that are currently lacking, the assistance value of w_i perceived by w_i for executing t is defined as

$$v_j(i-t) = \frac{\beta_1 \cdot \left(\left|S_{w_j} \bigcap \overline{S_t}\right| / \left|\overline{S_t}\right|\right) + \beta_2 \cdot \left(R_{w_j}\right) + \beta_3 \cdot c_i(\leftarrow j)}{\beta_4 \cdot \tau_{w_j} + \beta_5 \cdot d_{ij}}$$
(17)

where β_1 , β_2 , β_3 , β_4 , and β_5 are five parameters.

The task execution mechanism in a simplex social network can be designed based on the coordination tree, shown as Algorithm 3, whose time complexity is $O(|W|^2)$, where |W|is the number of workers. Then, $\forall w_i \in W_t$, w_i will execute Algorithm 3 to perform task *t*. Finally, the results of the different assigned workers will be combined, and the final result will be achieved by majority voting.

C. Task Execution in Multiplex Social Networks

Let the links among a crowd of workers in the multiplex social network be classified into λ different types $1, \ldots, \lambda$. We can set that each worker w_j has a threshold $\tau_{w_j}^k$ for the requests from *k*-type links, $1 \le k \le \lambda$. The threshold of one worker for another worker's request is determined by the maximum value of the minimum threshold for all social links of any worker within the path between the two workers. Therefore, we have the following definition.

Definition 9 (The Threshold of One Worker for the Request of Another Worker in the Multiplex Social Network): Let w_i be the assigned worker for task t, and let w_j be w_i 's contextual worker in the multiplex social network. The coordination path between w_i and w_j in the network is the path from w_i to w_j in the coordination tree of w_i for task t, which can be denoted **Algorithm 3:** Task Execution in a Simplex Social Network (t, w_i)

1 $b = 0; \overline{S_t} = S_t - S_{wi}; W_i(t) = \{w_i\};$ 2 Set the tags of all workers in the social network to 0; 3 **Create** Queue (Q); **Insert** Queue (Q, w_i); **Set** the tag of w_i to 1; 4 While ((!EmptyQueue (Q)) and (b == 0)) do: $w_{temp} =$ **Out** Queue(Q); 5 $\forall w_x \in N(w_{temp})$: /*Neighbors of w_{temp} */ 6 If b == 0: 7 8 $w_j = \arg \max_{\forall w_x \in N(w_{temp})} (v_x(i-t));$ 9 If the tag of w_i is 0: 10 **Insert** Queue (Q, w_i) ; Set the tag of w_j to 1; 11 12 If $\eta_1 \cdot m_{i \to j}(t) + \eta_2 \cdot c_{i \to j}(t) + \eta_3 \cdot c_i(\leftarrow j) + \eta_4 \cdot R_{w_i}$ $\geq \tau_{W_j}$ If $\overline{S_t} \cap S_{w_j} \neq \phi$: 13 $c_i(\leftarrow j) = c_i(\leftarrow j) - c_{i \to j}(t);$ 14 15 $c_i(\leftarrow i) = c_i(\leftarrow i) + c_{i \to j}(t);$ $\overline{S_t} = \overline{S_t} - S_{wj}; W_i(t) = W_i(t) \bigcup \{w_j\};$ 16 17 If $\overline{St} == \phi$: b = 1; 18 $\forall w_x \in W_i(t)$: cooperating to execute task t; 19 Output the executing result; 20 End.

as P_{ij} . Then, the threshold of w_i for w_i 's request is defined as

$$\tau_j(i) = \max_{\forall w_x \text{ in } P_{ij}} \left(\min_{1 \le k \le \lambda} \tau_{w_x}^k \right).$$
(18)

Based on Definition 8, the assistance value of w_j perceived by w_i for task *t* should be revised by considering the threshold defined in Definition 9, shown as the following:

$$Mv_{j}(i-t) = \left(\beta_{1} \cdot \left(\left|S_{w_{j}}\bigcap\overline{S_{t}}\right|/|\overline{S_{t}}|\right) + \beta_{2} \cdot \left(R_{w_{j}}\right) + \beta_{3} \cdot c_{i}(\leftarrow j)\right) \\ / \left(\beta_{4} \cdot \tau_{j}(i)\right).$$
(19)

We can now design the task execution process in a multiplex social network, shown as Algorithm 4, where w_i is the assigned worker for task *t*. $M(w_{\text{temp}})$ denotes the neighbors of all types of links of w_{temp} . The time complexity of Algorithm 4 is also $O(|W|^2)$. The process of Algorithm 4 is similar to the one of Algorithm 3, which is implemented based on the coordination tree.

VI. REWARD AFTER TASK EXECUTION

After the task is executed by the assigned workers and their assistant workers, the requester or the system will reward them according to the execution results. The rewards include monetary and reputation rewards.

A. Monetary Reward

Let the final result of t after majority voting be $\omega(t)$ and the executing result of t by w_i ($\forall w_i \in W_t$) be $\omega_i(t)$. We can then set two tolerance values to evaluate the accuracy of $\omega_i(t)$, Δ_1 and Δ_2 , $0 \le \Delta_1 \le \Delta_2$. If $|\omega_i(t) - \omega(t)| \le \Delta_1$, we can say that the accuracy of w_i 's executing task t is satisfactory, and w_i can now obtain the full monetary reward. If $\Delta_1 \le$ $|\omega_i(t) - \omega(t)| \le \Delta_2$, we can say that the accuracy of w_i 's executing task t is partially satisfactory, and w_i can now obtain

Algorithm 4: Task Execution in Multiplex Social Network (t, w_i)

1 $b = 0; \overline{S_t} = S_t - S_{wi}; W_i(t) = \{w_i\};$ 2 Set the tags of all workers in the social network to 0; 3 **Create** Queue (Q); **Insert** Queue (Q, w_i); **Set** the tag of w_i to 1; 4 While ((!EmptyQueue (Q)) and (b == 0)) do: 5 $w_{temp} =$ **Out** Queue(Q); $\forall w_x \in M(w_{temp})$: 6 If b == 0: 7 8 $\tau_{x-min} = \max$ value; For $(k=1, k \le \lambda, k++)$: 9 If (there is a k-type link between w_{temp} and w_x) and 10 $(\tau_{\chi}^{k} < \tau_{temp}): \tau_{\chi-min} = \tau_{\chi}^{k};$ If $w_{temp} == w_i$: $\tau_x(i) = \tau_{x-min}$; 11 **Else**: $\tau_x(i) = max(\tau_{temp}(i), \tau_{x-min});$ 12 $w_j = \arg \max_{\forall w_x \in M(w_{temp})} (Mv_x(i-t));$ 13 If the tag of w_i is 0: 14 **Insert** Queue (Q, w_i) ; 15 16 Set the tag of w_i to 1; If 17 $\eta_1 \cdot m_{i \to j}(t) + \eta_2 \cdot c_{i \to j}(t) + \eta_3 \cdot c_i(\leftarrow j) + \eta_4 \cdot R_{w_i} \ge \tau_j(i):$ If $\overline{S_t} \cap S_{w_i} \neq \phi$: 18 $c_i(\leftarrow j) = c_i(\leftarrow j) - c_{i \rightarrow j}(t); c_j(\leftarrow i) =$ 19 $c_i(\leftarrow i) + c_{i\rightarrow j}(t);$ $\overline{St} = \overline{St} - S_{wj}; W_i(t) = W_i(t) \bigcup \{w_j\}; \}$ 20 If $\overline{S_t} == \phi : b = 1;$ 21 22 $\forall w_x \in W_i(t)$: cooperating to execute task t;

22 $\forall w_x \in w_i(t)$: cooperating to execute task 23 **Output** the executing result; 24 **End**.

Algorithm 5: Monetary Reward Mechanism (t)

1 $W_t(ok) = \{\}; //Workers providing satisfactory accuracy$ 2 $\forall w_i \in W_t$: 3 If $|\omega_i(t) - \omega(t)| \leq \Delta_1$: $\{M_i = \gamma_{wi}; b_t = b_t - M_i; W_t(ok) = W_t(ok) \bigcup \{w_i\}\};\$ 4 If $\Delta_1 \leq |\omega_i(t) - \omega(t)| \leq \Delta_2$: 5 $\{M_i = \Delta_1 / |\omega_i(t) - \omega(t)| \cdot \gamma_{wi}; b_t = b_t - M_i\};\$ 6 If $\Delta_2 \leq |\omega_i(t) - \omega(t)|$: $M_i = 0$; 8 $\forall w_i \in W_t(\text{ok}): M_i = M_i + b_t / |W_t(\text{ok})|;$ 9 $\forall w_i \in W_t$: Reward $(1 - \lambda)M_i$ to w_i ; 10 $\forall w_i \in (W_i(t) - \{w_i\})$: Reward $\lambda M_i \cdot (|S_i^t| / |S_t - S_{wi}|)$ 11 12 End.

part of the monetary reward; if $\Delta_2 \leq |\omega_i(t) - \omega(t)|$, we can say that the accuracy of w_i 's executing task t is unsatisfactory, and w_i cannot obtain a monetary reward. Moreover, finally there may be an unspent part of the budget of t, which can be used as a bonus to reward the assigned workers that can achieve satisfactory accuracy.

After w_i receives the monetary reward, M_i , he will distribute some part of M_i to the assistant workers. We can set a parameter λ , $0 \le \lambda \le 1$, that denotes the percentage of monetary reward that w_i is willing to distribute to the assistant workers, i.e., w_i will distribute λM_i to $\forall w_i \in (W_i(t) - \{w_i\})$.

Let the reservation wage of w_i for task t be γ_{wi} . γ_{wi} is also thought of as the full monetary reward received by w_i from the system for successfully executing task t. We can now set the monetary reward mechanism as Algorithm 5, in which S_j^t denotes the real skills contributed by w_i for task t.

Algorithm 6: Reputation Reward Mechanism (t)

_			
1	$\forall w_i \in W_t$:		
2	If $ \omega_i(t) - \omega(t) \leq \Delta_1$:		
3	$R_{wi} = R_{wi} + \mu \zeta_t;$		
4	$\forall w_j \in (W_i(t) - \{w_i\}): R_{wj} = R_{wj} + (1 - \mu)\zeta_t;$		
5	If $\Delta_1 \leq \omega_i(t) - \omega(t) \leq \Delta_2$:		
6	$R_{wj} = R_{wj} + \mu \cdot \Delta_1 / \omega_i(t) - \omega(t) \cdot \zeta_t;$		
7	$\forall w_j \in (W_i(t) - \{w_i\}):$		
8			
9	If $\Delta_2 \leq \omega_i(t) - \omega(t) $:		
10	$R_{wj} = R_{wj} - \mu \zeta_t;$		
11	$\forall w_{j} \in (W_{i}(t) - \{w_{i}\}): R_{wj} = R_{wj} - (1 - \mu)\zeta_{t};$		
12 Fnd			

B. Reputation Reward

We first set a value ζ_t for the reputation reward of task t; moreover, we also set another value μ that measures the relative obligation incurred by the assigned worker to the assistant workers for executing the task. Generally, we can set $0.5 << \mu \le 1$; this means that the assigned worker assume a primary obligation to complete the task successfully and the assistant workers only undertake the secondary obligation. The reputation reward mechanism can then be designed as Algorithm 6.

VII. EXPERIMENTAL VALIDATION AND ANALYSES

A. Experimental Setting

The experiments are conducted using a real-world dataset extracted from Freelancer.¹ The dataset includes information on the workers and information on the tasks. Specifically, the information for each worker includes the set of skills he has and the reserved wage he declares; and for each task, the information includes the required skills and the budget presented by the requester. At Freelancer, each worker can list at most five different skills in his profile, and each requester can also input at most five different required skills in his profile.

The original collected data include 9642 tasks and 997 workers. The tasks require 644 different skills in total, but the workers only possess 107 different skills. Thus, many tasks cannot be completed by these workers. To allow every task a chance to be completed, we remove the tasks that require skills that are not possessed by any of the workers. The final dataset contains 697 tasks and 997 workers. Fig. 2 shows the distribution of the tasks' required skills, the workers' skills, the tasks' budgets, and the workers' reservation wages; the y-axis denotes the occurrence frequencies of the varying numbers of the four factors, respectively, within the collected data. From Fig. 2, we can see that both the tasks and the workers' skills follow the power law distribution; this indicates that some skills are very popular whereas others are not. If a task requires some unpopular skills, it may be difficult for the task to be matched with perfectly suitable workers. The task budgets also follow the power law distribution, whereas the workers' reservation wages follow a normal distribution. Here, it is noted that the average budget is much higher than

the average reserved wage. Thus, it is possible that a task can be allocated to several workers.

Some recent notable studies have shown that workers are often connected through social networks [12], [13]. Moreover, in some popular crowdsourcing platforms, workers can register using their Facebook or LinkedIn accounts, which can include their personal information and social connections. Therefore, in the experiments, we use Facebook's social network to model the workers' social networks, i.e., we combine the Facebook network with the Freelancer dataset to construct a crowdsourcing platform with a social network structure. We randomly extract 997 nodes from the downloaded Facebook network, and we find that these nodes are connected. The average degree of the network is 24.93. The height of the breadth-first traversal tree starting from any node is 4.6, so it is not difficult to find coordination workers for each assigned worker in the network.

In the experiments, for simplification, all the factors are considered equally. Therefore, we set all parameters, including α_1 , α_2 , α_3 , β_1 , β_2 , β_3 , β_4 , β_5 , η_1 , η_2 , η_3 , and η_4 , to be normalized 1. If the parameters are not normalized, some factors would be overwhelmed by other factors since the value scopes of varying factors are very different. The distribution of workers' accuracies follows a normal distribution N(0.7, 0.1); moreover, the accuracy values are set within the range [0, 1].

Our experiments are implemented with MATLAB R2014b on a PC with an Intel Dual Core 3.40 GHz CPU and 16 GB memory.

B. Benchmark Approaches

In the experiments, we compare our presented contextaware task allocation approach with two previous benchmark approaches: 1) the straightforward task allocation approach and 2) the decomposition-based task allocation approach.

The *straightforward task allocation approach* is a traditional approach that has been used in previous crowdsourcing systems [30]. In this approach, the requester (or the crowdsourcing system) allocates the task to a worker who fully satisfies the skill requirements of the task. Therefore, each assigned worker can perform the task individually and independently. If the assigned worker's required wage does not exceed the budget, the task can be redundantly assigned to other workers until the budget is used up. If the requester (or the system) cannot find a worker who fully satisfies the skill requirements of the task, the task cannot be allocated successfully.

The *decomposition-based task allocation approach* [2] is a popular approach for performing complex tasks in which each complex task is decomposed into a flow of simple subtasks; the subtasks are then allocated to workers, each of whom can fully satisfy the skill requirements of the assigned subtask and perform the assigned subtask independently. If all of the subtasks are allocated successfully, the original complex task is deemed to have been allocated successfully. If the budget is not used up, the complex task will be redundantly allocated multiple times.

C. Validation of the Task Allocation Efficiency

In the task allocation objective defined in (1) and (4), a task will be redundantly assigned to as many workers as possible



Fig. 2. Statistics of the collected data at Freelancer website. (a) Distribution of tasks' skills. (b) Distribution of workers' skills. (c) Distribution of tasks' budgets. (d) Distribution of workers' reserved wages.



Fig. 3. Number of successful allocations made using the three approaches under varying situations. (a) Increasing the budget. (b) Increasing the skill number of a task. (c) Increasing the worker quantity. (d) Increasing the task quantity.

under a given budget, which can improve the accuracy of the solution [2]. Thus, one important task allocation objective is to maximize the redundancy degree of task allocation. Therefore, we can use the number of successful allocations to measure the allocation efficiencies of the three approaches.

First, we test the number of successful redundant allocations when the budget is varied. In the test, we enter the tasks into the system one by one and then record the average allocation number for all tasks. Fig. 3(a) shows that with the same budget, the average successful allocation number for all tasks in our approach is much higher than the ones obtained using the other two approaches; moreover, the average successful allocation number of our approach increases with the increase of the budget more drastically than do the other two benchmark approaches. The reason is: when the task's budget increases, our approach allows each assigned worker to find more appropriate assistant workers, thus increasing the number of successful allocations (here, a successful allocation means that an assigned worker and his assistant workers can satisfy all of the skill requirements of the task). In comparison, since the reserved wage of each worker is much lower than the task's budget, the possible workers who can fully satisfy the skill requirements of the task can be found whether the budget is low or high; therefore, an increase in the budget has no obvious effect in the other two benchmark approaches.

Second, we test the number of successful redundant allocations when the number of skills required by a task increases, shown as Fig. 3(b). At first when a task requires few skills, our approach cannot achieve better performance because the budget is quickly used up with our approach. However, when the tasks require more and more skills, the other two benchmark approaches cannot find appropriate workers who can fully satisfy the skill requirements of the tasks. Thus, their successful allocation numbers deteriorate more drastically than does our approach. Moreover, when the number of skills needed for the task exceeds the system's specified number of skills, i.e., five skills, our approach can still allocate the task quite successfully since it can utilize the contextual workers' skills, whereas the decomposition-based task allocation can only allocate very few times, and the straightforward task allocation cannot find any workers who can fully satisfy the need for more than five skills.

Third, we test the number of successful redundant allocations when the number of workers in the crowd increases, shown as Fig. 3(c). We can see that the allocation number of our approach increases more rapidly than in the other two benchmark approaches. The reason is that our model is able to harness the power of more workers when a single worker cannot complete the task individually; in comparison, other two benchmark approaches can only utilize the power of the workers who can fully satisfy the skill requirements of the task.

Fourth, to consider the real-world situation in which requesters may outsource numerous tasks simultaneously, we test the throughput of the three approaches. In the experiment, we assume that each task is only allocated one time and that a worker cannot be assigned to other tasks if that worker has already been assigned to a task. We then test the ratio of successful allocated tasks to all tasks, shown as Fig. 3(d). We can see that our approach performs better than the other two benchmark approaches. The reason is that our approach allows



Fig. 4. Task execution accuracy of the three approaches under varying situations. (a) Increasing the budget. (b) Increasing the skill number of a task. (c) Increasing the worker quantity.

more workers to be considered in the task allocation even the workers themselves cannot fully meet the skill requirements of the tasks; in comparison, the other two benchmark approaches only consider the workers who can fully satisfy the skill requirements of the tasks; thus, the number of qualified workers may be fewer than in our approach.

D. Validation of the Task Execution Efficiency

We now compare the task executing accuracy in the three approaches. In the experiment, each worker w_i has a random executing accuracy a_{wi} ; the distribution of workers' accuracies follows a normal distribution N(0.7, 0.1). Moreover, the accuracy values are set within the range [0, 1].

After a task is allocated redundantly, the task will be executed more than one time. The accuracy of one execution of a task t (e.g., the *j*th execution), $a_i(t)$, is determined by the average accuracy of the workers participating in the execution. In the straightforward approach, there is only one worker in each execution; thus, the accuracy of each execution is the accuracy of the assigned worker, $a_i(t) = a_{wi}$. In the decomposition-based task allocation, let the task be decomposed into m subtasks and let only one worker participate in one execution of each subtask; thus, there are *m* workers participating in one execution of the entire task. In our presented approach, let there be *m* workers participating in one execution of the task; thus, there are one assigned worker and m-1assistant workers. Therefore, the accuracy of one execution of a task in the decomposition-based task allocation approach and our presented approach is

$$a_j(t) = \frac{1}{m} \sum_{i=1}^m a_{w_i}.$$
 (20)

Since each task is allocated redundantly multiple times (n), the overall accuracy of the task is

$$a(t) = 1 - \prod_{j=1}^{n} (1 - a_j(t)).$$
(21)

Intuitively, if a task is executed more times, the result will be more accurate. The experimental results are shown in Fig. 4. Fig. 4(a) shows that the executing accuracy with our approach increases quite rapidly with the increase of the

task's budget, finally reaching 1. Fig. 4(b) shows the execution accuracy when we increase the number of the task's skills; we can see that the execution accuracy of our approach remains very close to 1 when the number of required skills is less than 13, but the execution accuracy in other two approaches deteriorates drastically as the number of required skills increases. Fig. 4(c) shows the execution accuracy when the worker quantity increases. Compared with the other two benchmark approaches, our presented approach achieves much higher execution accuracy even when the quantity of workers is small.

Therefore, our presented approach achieves better performance in task execution efficiency than the other two benchmark approaches. The reason is similar to that described in Section VII-C; for brevity, a detailed description is omitted.

E. Validation of the Reputation Mechanism

We now test the effects of our reputation mechanism when there are unreliable or malicious workers. In particular, we compare the task execution accuracies when the reputation mechanism is adopted and when it is not. In the experiments, we randomly set some workers as unreliable. For simplicity, we assume that the results of task execution are binary values. We can then adopt a method similar to that described in [33] in which the unreliable workers also contribute in some degree to the accuracy of the task and hence, even if all workers are unreliable, the task can still achieve some accuracy if the reputation mechanism is adopted. The unreliable workers' accuracies follow the normal distribution N(0.2, 0.1). When the reputation mechanism is adopted, the system gradually identifies the unreliable workers and then utilizes the results generated by the unreliable workers. The results are shown in Fig. 5, in which z-axis shows the differences of task execution when the reputation mechanism is or is not adopted in a variety of situations.

Fig. 5(a) shows the results for varying task budgets (x-axis) and proportions of unreliable workers (y-axis). We can see that when there are more unreliable workers, the model adopting a reputation mechanism performs much better than the model that does not; therefore, our presented reputation mechanism can effectively address the presence of unreliable workers. Moreover, we can also see that the reputation mechanism



Fig. 5. Differences in task execution accuracies when a reputation mechanism for different ratios of unreliable workers is or is not adopted under varying situations. (a) Varying the budgets. (b) Varying the task skills. (c) Varying the worker quantity.



Fig. 6. Average number of workers allocated to a task in two types of networks under varying situations. (a) Varying budgets. (b) Varying worker quantity.



Fig. 7. Tests comparing the two task allocation methods in multiplex networks. (a) Test on the number of successful allocations. (b) Test on the execution accuracy. (c) Test on communication costs among workers.

can provide more obvious benefits when the task's budget is insufficient.

Fig. 5(b) shows the results for varying numbers of task skills (x-axis) and proportions of unreliable workers (y-axis). When a task requires more skills, i.e., the task is more complex, the reputation mechanism performs much better than the model that does not incorporate a reputation mechanism. Therefore, our presented reputation mechanism can effectively address the presence of the unreliable workers, especially when the task is complex.

Fig. 5(c) shows the results for varying numbers of workers (x-axis) and proportions of unreliable workers (y-axis). The results show that the reputation mechanism can perform better especially when the workers are not sufficient.

F. Comparing the Simplex and Multiplex Networks

We now test our approach in simplex and multiplex networks. Here, we consider how many workers would be involved in a single round of execution of a task. In Fig. 6, the y-axis denotes the average number of workers assigned to a task in each allocation. We can see in Fig. 6(a) and (b) that the average number of involved workers in a multiplex network is often more than that in a simplex network. This means that simplex networks often make it easier to find appropriate workers to satisfy the task objective.

Fig. 6(a) shows the number of participating workers increases with the increase of budgets. The likely reason is that the workers whose skills are close to the skills required by a task will be preferably assigned while the budget is limited, but some other workers whose skills are not close to the skills required by the task may also be assigned when the budget is higher. Fig. 6(b) shows that there is no obvious variation tendency between the number of participating workers and the increase of worker quantity. The reason is that the workers are randomly added to the crowd.

G. Comparing the Two Task Allocation Methods in Multiplex Networks

We now compare the two task allocation methods in multiplex networks: 1) the worker-oriented method that uses Algorithm 1 and 2) the network layer and workeroriented method that uses Algorithm 2. The following three performance indices are tested under varying task quantities: 1) number of successful allocations; 2) execution accuracy; and 3) communication costs among the participating workers.

The experimental results in Fig. 7(a)–(c) show that the worker-oriented method achieves better performance on number of successful allocations and task execution accuracy than the network layer and worker-oriented method. The reason is that the former method seeks the appropriate workers globally from the whole network, making it more likely that optimal workers will be found. However, the latter method can achieve better performance than the former on communication costs among participating workers, because the latter method seeks assistant workers preferably within the network layer, thereby reducing the communication costs.

VIII. CONCLUSION

This paper aims to solve two typical problems noted in the previous studies of crowdsourcing complex tasks: 1) the requesters undertake a heavy burden when decomposing complex tasks into a set of micro-subtasks and 2) the reliability may not be ensured when there are many malicious workers in the crowd. By considering a current general situation in which the workers are often connected by social networks, this paper explores a context-aware reliable crowdsourcing approach that can solve the above two problems in the social network environments.

This paper implements the approach by defining a reasonable concept of crowdsourcing value that can be used to measure the probability of a worker's being assigned a task when the context of the worker in the social network is considered. The approach addresses two typical social networks: 1) simplex networks and 2) multiplex networks. The experiments on a real-world dataset show that the presented approach outperforms previous benchmark approaches with respect to task allocation and execution efficiencies; moreover, the presented approach can effectively address the situation in which there are many unreliable workers.

In this paper, the social networks among workers are fixed. In reality, social networks may sometimes be dynamic. Therefore, in the future, we will explore the adaptive mechanism of our approach to dynamic social networks.

REFERENCES

- [1] J. Howe, "The rise of crowdsourcing," *Wired Mag.*, vol. 14, no. 6, pp. 1–4, 2006.
- [2] L. Tran-Thanh, T. D. Huynh, A. Rosenfeld, S. D. Ramchurn, and N. R. Jennings, "BudgetFix: Budget limited crowdsourcing for interdependent task allocation with quality guarantees," in *Proc. 13th Int. Conf. Auton. Agents Multiagent Syst. (AAMAS)*, Paris, France, May 2014, pp. 477–484.
- [3] N. Luz, N. Silva, and P. Novais, "A survey of task-oriented crowdsourcing," Artif. Intell. Rev., vol. 44, no. 2, pp. 187–213, 2015.

- [4] D. R. Karger, S. Oh, and D. Shah, "Budget-optimal task allocation for reliable crowdsourcing systems," *Oper. Res.*, vol. 62, no. 1, pp. 1–24, 2014.
- [5] M. Venanzi, A. Rogers, and N. R. Jennings, "Trust-based fusion of untrustworthy information in crowdsourcing applications," in *Proc. 12th Int. Conf. Auton. Agents Multiagent Syst. (AAMAS)*, St. Paul, MN, USA, May 2013, pp. 829–836.
- [6] S. Assadi, J. Hsu, and S. Jabbari, "Online assignment of heterogeneous tasks in crowdsourcing markets," in *Proc. 3rd AAAI Conf. Human Comput. Crowdsourcing (HCOMP)*, San Diego, CA, USA, Nov. 2015, pp. 12–21.
- [7] J. Song, S. Lee, and J. Kim, "CrowdTarget: Target-based detection of crowdturfing in online social networks," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, Denver, CO, USA, Oct. 2015, pp. 793–804.
- [8] L. de Alfaro, V. Polychronopoulos, and M. Shavlovsky, "Reliable aggregation of Boolean crowdsourced tasks," in *Proc. 3rd AAAI Conf. Human Comput. Crowdsourcing (HCOMP)*, San Diego, CA, USA, Nov. 2015, pp. 42–51.
- [9] Y. Zhang and M. van der Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *Proc. 31st Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Orlando, FL, USA, Mar. 2012, pp. 2140–2148.
- [10] S. Penmatsa and A. T. Chronopoulos, "Cost minimization in utility computing systems," *Concurrency Comput. Pract. Exp.*, vol. 26, no. 1, pp. 287–307, 2014.
- [11] J. Chamberlain, "Groupsourcing: Distributed problem solving using social networks," in *Proc. 2nd AAAI Conf. Human Comput. Crowdsourcing (HCOMP)*, Pittsburgh, PA, USA, Nov. 2014, pp. 22–29.
- [12] M. Yin, M. L. Gray, S. Suri, and J. W. Vaughan, "The communication network within the crowd," in *Proc. 25th Int. World Wide Web Conf.* (WWW), Montreal, QC, Canada, Apr. 2016, pp. 1293–1303.
- [13] M. L. Gray, S. Suri, S. S. Ali, and D. Kulkarni, "The crowd is a collaborative network," in *Proc. 19th ACM Conf. Comput. Supported Cooperative Work Soc. Comput. (CSCW)*, San Francisco, CA, USA, Feb./Mar. 2016, pp. 134–147.
- [14] H. Gao, G. Barbier, and R. Goolsby, "Harnessing the crowdsourcing power of social media for disaster relief," *IEEE Intell. Syst.*, vol. 26, no. 3, pp. 10–14, May/Jun. 2011.
- [15] S. L. Lim, D. Quercia, and A. Finkelstein, "StakeSource: Harnessing the power of crowdsourcing and social networks in stakeholder analysis," in *Proc. 32nd ACM/IEEE Int. Conf. Softw. Eng. (ICSE)*, Cape Town, South Africa, May 2010, pp. 239–242.
- [16] C. Gao and J. Liu, "Network-based modeling for characterizing human collective behaviors during extreme events," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 1, pp. 171–183, Jan. 2017.
- [17] A. Bozzon, M. Brambilla, S. Ceri, M. Silverstri, and G. Vesci, "Choosing the right crowd: Expert finding in social networks," in *Proc. 16th Int. Conf. Extending Database Technol. 16th Int. Conf. Database Theory* (*EDBT/ICDT*), Genoa, Italy, Mar. 2013, pp. 637–648.
- [18] Q. Liu, T. Luo, R. Tang, and S. Bressan, "An efficient and truthful pricing mechanism for team formation in crowdsourcing markets," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., Jun. 2015, pp. 567–572.
- [19] S. He, D.-H. Shin, J. Zhang, and J. Chen, "Toward optimal allocation of location dependent tasks in crowdsensing," in *Proc. 33rd Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Toronto, ON, Canada, Apr./May 2014, pp. 745–753.
- [20] J. J. Horton and L. B. Chilton, "The labor economics of paid crowdsourcing," in *Proc. 11th ACM Conf. Electron. Commerce (EC)*, Cambridge, MA, USA, Jun. 2010, pp. 209–218.
- [21] P. Dai, C. H. Lin, Mausam, and D. S. Weld, "POMDP-based control of workflows for crowdsourcing," *Artif. Intell.*, vol. 202, no. 1, pp. 52–85, 2013.
- [22] M. Kargar, A. An, and M. Zihayat, "Efficient bi-objective team formation in social networks," in *Proc. Eur. Conf. Mach. Learn. Principles Pract. Knowl. Disc. Databases (ECML PKDD)*, Bristol, U.K., Sep. 2012, pp. 483–498.
- [23] A. Tamilin, I. Carreras, E. Ssebaggala, A. Opira, and N. Conci, "Contextaware mobile crowdsourcing," in *Proc. 14th ACM Int. Conf. Ubiquitous Comput. (UbiComp)*, Pittsburgh, PA, USA, Sep. 2012, pp. 717–720.
- [24] Y. Jiang, Y. Zhou, and W. Wang, "Task allocation for undependable multiagent systems in social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 8, pp. 1671–1681, Aug. 2013.
- [25] M. Kargar and A. An, "Discovering top-k teams of experts with/without a leader in social networks," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manag. (CIKM)*, Glasgow, U.K., Oct. 2011, pp. 985–994.

- [26] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min. (KDD)*, Paris, France, Jun./Jul. 2009, pp. 467–476.
- [27] Y. Jiang, Y. Zhou, and Y. Li, "Reliable task allocation with load balancing in multiplex networks," ACM Trans. Auton. Adapt. Syst., vol. 10, no. 1, 2015, Art. no. 3.
- [28] Y. Zeng, X. Chen, Y.-S. Ong, J. Tang, and Y. Xiang, "Structured memetic automation for online human-like social behavior learning," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 102–115, Feb. 2017.
- [29] P. C. Pinto, P. Thiran, and M. Vetterli, "Locating the source of diffusion in large-scale networks," *Phys. Rev. Lett.*, vol. 109, no. 6, 2012, Art. no. 068702.
- [30] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with mechanical turk," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.* (*CHI*), Florence, Italy, Apr. 2008, pp. 453–456.
- [31] C. Wang, L. Cao, and C.-H. Chi, "Formalization and verification of group behavior interactions," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 8, pp. 1109–1124, Aug. 2015.
- [32] E. Franchi, A. Poggi, and M. Tomaiuolo, "Social media for online collaboration in firms and organizations," *Int. J. Inf. Syst. Model. Design*, vol. 7, no. 1, pp. 18–31, 2016.
- [33] A. Dasgupta and A. Ghosh, "Crowdsourced judgement elicitation with endogenous proficiency," *Proc. 22nd Int. Conf. World Wide Web (WWW)*, Rio de Janeiro, Brazil, May 2013, pp. 319–330.
- [34] T. Hoßfeld *et al.*, "Survey of Web-based crowdsourcing frameworks for subjective quality assessment," in *Proc. 16th Int. Workshop Multimedia Signal Process. (MMSP)*, Jakarta, Indonesia, Sep. 2014, pp. 1–6.
- [35] Y. Kong, M. Zhang, and D. Ye, "A belief propagation-based method for task allocation in open and dynamic cloud environments," *Knowl. Based Syst.*, vol. 115, pp. 123–132, Jan. 2017.
- [36] D. Ye, M. Zhang, and A. V. Vasilakos, "A survey of self-organization mechanisms in multiagent systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 3, pp. 441–461, Mar. 2017.
- [37] R. Rana, C. T. Chou, N. Bulusu, S. Kanhere, and W. Hu, "Ear-phone: A context-aware noise mapping using smart phones," *Pervasive Mobile Comput.*, vol. 17, Part A, pp. 1–22, Feb. 2015.
- [38] X. Hu, X. Li, E. Ngai, V. Leung, and P. Kruchten, "Multidimensional context-aware social network architecture for mobile crowdsensing," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 78–87, Jun. 2014.
- [39] F. Alt, A. S. Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz, "Location-based crowdsourcing: Extending crowdsourcing to the real world," in *Proc. 6th Nordic Conf. Human Comput. Interact. Extending Boundaries (NordiCHI)*, Reykjavik, Iceland, Oct. 2010, pp. 13–22.
- [40] J. Ren, Y. Zhang, K. Zhang, and X. S. Shen, "SACRM: Social aware crowdsourcing with reputation management in mobile sensing," *Comput. Commun.*, vol. 65, no. 1, pp. 55–65, Jul. 2015.
- [41] J. Ren, Y. Zhang, K. Zhang, and X. S. Shen, "Exploiting mobile crowdsourcing for pervasive cloud services: Challenges and solutions," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 98–105, Mar. 2015.

Bo An received the Ph.D. degree in computer science from the University of Massachusetts, Amherst, MA, USA, in 2011.

He is currently a Nanyang Assistant Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. From 2010 to 2012, he was a Postdoctoral Researcher with the University of Southern California, Los Angeles, CA, USA. He has published over 70 referred papers at top conferences, such as AAMAS, IJCAI, AAAI,

ICAPS, and KDD, and journals such as ACM/IEEE TRANSACTIONS, Artificial Intelligence Journal, and the Journal of Autonomous Agents and Multi-Agent Systems. His current research interests include artificial intelligence, multiagent systems, game theory, crowdsourcing, and automated negotiation.

Dr. An was a recipient of the 2010 IFAAMAS Victor Lesser Distinguished Dissertation Award, the Operational Excellence Award from the Commander, the First Coast Guard District of the United States, the Best Innovative Application Paper Award at AAMAS-12, the 2012 INFORMS Daniel H. Wagner Prize for Excellence in Operations Research Practice, and the Innovative Application Award at IAAI-16. He was invited to give an Early Career Spotlight talk at IJCAI'17. He led the team HogRider, which won the 2017 Microsoft Collaborative AI Challenge. He is a member of the Board of Directors of IFAAMAS, the Associate Editor of the *Journal of Autonomous Agents and Multi-Agent Systems*, and a member of the editorial board of JAIR. He has served as the program committee members for many top conferences, such as AAMAS, IJCAI, and AAAI and was the cochair for several key international conferences, including the Tutorial Cochair of IJCAI-18.



Yichuan Jiang (SM'13) received the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2005.

He is currently a Full Professor with the School of Computer Science and Engineering, Southeast University, Nanjing, China. He has published over 80 scientific articles in refereed journals and conference proceedings, such as the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the

IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, the IEEE TRANSACTIONS ON CYBERNETICS, the ACM Transactions on Autonomous and Adaptive Systems, the Journal of Autonomous Agents and Multi-Agent Systems, the Journal of Parallel and Distributed Computing, IJCAI, AAAI, AAMAS, and ECAI. His current research interests include multiagent systems, crowdsourcing, and social networks.

Dr. Jiang was a recipient of the Best Paper Award from PRIMA06 and the Best Student Paper Awards (twice) from ICTAI13 and ICTAI14.



Jiuchuan Jiang received the M.E. degree in computer science from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2009.

He is currently with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He has published several scientific articles in refereed journals and conference proceedings, such as the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND

CYBERNETICS: SYSTEMS, and the International Conference on Autonomous Agents and Multiagent Systems. His current research interests include crowdsourcing, multiagent systems, and social networks.



Donghui Lin received the Ph.D. degree in computer science from Kyoto University, Kyoto, Japan, in 2008.

He is currently an Assistant Professor with the Department of Social Informatics, Kyoto University. He has published over 40 scientific papers in refereed journals and conference proceedings. His current research interests include services computing, crowdsourcing, and multiagent systems.

Dr. Lin was a recipient of the 2012 Achievement

Award of the Institute of Electronics, Information and Communication Engineers. He served as the program cochair or general cochair at several key international conferences and workshops in the area of services computing and services science.