Understanding Crowdsourcing Systems from a Multiagent Perspective and Approach

JIUCHUAN JIANG and BO AN, Nanyang Technological University
YICHUAN JIANG, Foshan University; Southeast University
DONGHUI LIN, Kyoto University
ZHAN BU and JIE CAO, Nanjing University of Finance and Economics
ZHIFENG HAO, Foshan University

Crowdsourcing has recently been significantly explored. Although related surveys have been conducted regarding this subject, each has mainly consisted of a review of a single aspect of crowdsourcing systems or on the application of crowdsourcing in a specific application domain. A crowdsourcing system is a comprehensive set of multiple entities, including various elements and processes. Multiagent computing has already been widely envisioned as a powerful paradigm for modeling autonomous multi-entity systems with adaptation to dynamic environments. Therefore, this article presents a novel multiagent perspective and approach to understanding crowdsourcing systems, which can be used to correlate the research on crowdsourcing and multiagent systems and inspire possible interdisciplinary research between the two areas. This article mainly discusses the following two aspects: (1) The multiagent perspective can be used for conducting a comprehensive survey on the state of the art of crowdsourcing, and (2) the multiagent approach can bring about concrete enhancements for crowdsourcing technology and inspire future research directions that enable crowdsourcing research to overcome the typical challenges in crowdsourcing technology. Finally, this article discusses the advantages and disadvantages of the multiagent perspective by comparing it with two other popular perspectives on crowdsourcing: the business perspective and the technical perspective.

CCS Concepts: • **Computing methodologies → Multi-agent systems**; • **Information systems → Crowdsourcing**; • **Human-centered computing → Collaborative and social computing**;

Additional Key Words and Phrases: Crowdsourcing systems, multiagent systems, crowdsourcing elements, crowdsourcing processes

Y. Jiang and Z. Hao were supported by the National Natural Science Foundation of China (61472079, 61170164, and 61472089), the Natural Science Foundation of Jiangsu Province of China (BK20171363), the Joint Fund of the National Natural Science Foundation of China and Guangdong Province (U1501254), the Science and Technology Planning Project of Guangdong Province (2015B010131015 and 2015B010108006), the Natural Science Foundation of Guangdong Province (2014A030308008), and Guangdong Regular University International and Hong Kong, Macao, and Taiwan Cooperative Innovation Platform and International Cooperation Major Projects (2015KGJHZ023).
Authors' addresses: J. Jiang and B. An, School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore; emails: jiangjiuchuan@163.com, boan@ntu.edu.sg; Y. Jiang, School of Mathematics and Big Data, Foshan University, Foshan 528000, China, and School of Computer Science and Engineering, Southeast University, Nanjing 211189, China; email: yjiang@seu.edu.cn; D. Lin, Department of Social Informatics, Kyoto University, Kyoto, Japan; email: lindh@i.kyoto-u.ac.jp; Z. Bu and J. Cao, Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing 210003, China; emails: buzhan@nuaa.edu.cn, caojie690929@163.com; Z. Hao, School of Mathematics and Big Data, Foshan University, Foshan 528000, China; email: zfhao@fosu.edu.cn.

https://doi.org/10.1145/3226028

ACM Reference format:
Jiuchuan Jiang, Bo An, Yichuan Jiang, Donghui Lin, Zhan Bu, Jie Cao, and Zhifeng Hao. 2018. Understanding Crowdsourcing Systems from a Multiagent Perspective and Approach. *ACM Trans. Auton. Adapt. Syst.* 13, 2, Article 8 (July 2018), 32 pages.
https://doi.org/10.1145/3226028

1 INTRODUCTION

Crowdsourcing is a new task allocation concept wherein a task can be outsourced to workers who are chosen from a crowd instead of being performed by a designated agent [1], which is often suitable for tasks that are trivial for humans but difficult for computers [2, 3], such as classification tasks [5]. Compared with traditional task allocation, the advantages of crowdsourcing include the following: faster completion speed, lower costs, higher accuracy, and completion of tasks that computers cannot perform.

Because crowdsourcing has broad application prospects and significant business value, many studies have been conducted in recent years. Of the great progress in this area and the wide variety of research results, several related surveys are presented to provide a taxonomy and review of this subject. However, most previous surveys have mainly reviewed a single aspect of crowdsourcing or the application of crowdsourcing in one specific domain, such as the survey of tasks in crowdsourcing [6, 7], the survey of crowdsourcing for the data mining domain [8], the survey of crowdsourcing on the world-wide-web [9], the survey of the future of crowdsourcing [122], the survey of the difference between crowdsourcing and human computation [123], and the survey of crowdsourcing in software engineering [124]. Although a few surveys have attempted to present a more general review of crowdsourcing [4], they have only reviewed the definitions of crowdsourcing and typical crowdsourcing systems. Moreover, they did not perform a systematic review of a comprehensive set of aspects of crowdsourcing systems and did not consider the interactions among various elements in crowdsourcing systems.

To address the above problems in the previous related surveys, this article aims at providing a general and macroscopic review of various elements and processes of crowdsourcing systems. In general, each crowdsourcing application includes the following key elements: (1) *tasks*, which are the outsourced objects that have various characteristics in reality, among which micro-tasks and complex tasks are two typical types [6]; (2) *requesters*, who release the tasks, decompose complex tasks, and evaluate the solutions for the tasks; (3) *system platforms*, which provide efficient measures that can manage and organize the entire crowdsourcing process and undertake some affairs of requesters; and (4) *workers*, i.e., the crowd of people, who possess various skills and can provide computational power for performing the outsourced tasks. When a crowdsourcing system wants to perform a task, the following processes are necessary from the time the task comes to the system to the time the task is completely finished: (1) the *pre-execution process*, in which the task is processed by the requester or system platform before it is executed by workers, which includes task analysis, task decomposition, and task allocation; (2) the *execution process*, which represents the execution of the task by the assigned workers; and (3) the *post-execution process*, which represents the affairs after the execution of the task and often includes aspects such as aggregation and quality control of the results and payment of the workers.

Crowdsourcing systems are often autonomous and dynamic, in which the requesters and workers often make decisions autonomously and the crowd of workers and the workloads are continuously changing. An important promising trend of crowdsourcing is an autonomous crowdsourcing service with the capability of autonomous adaptation in highly dynamic social environments [9, 10, 11, 125]. Because multiagent computing has already been widely envisioned to be a powerful paradigm for modeling autonomous multi-entity systems with adaptation for dynamic environments, this article presents a multiagent perspective and approach to understanding crowdsourcing systems. We present the following two main aspects:

1) The multiagent perspective can be used for conducting a comprehensive survey on the state of the art of crowdsourcing. For instance, the terminology, definitions, and classification that are used in the related multiagent system literature can be used to understand, analyze, and classify the state of the art of various elements and processes of crowdsourcing systems.
2) The multiagent approach can bring about concrete enhancements for crowdsourcing technology and inspire concrete future research directions that enable crowdsourcing research to overcome the typical challenges in crowdsourcing technology. The reason is that the majority of existing typical challenges in concrete crowdsourcing technology (e.g., task allocation, task decomposition and coordination, and designing mechanisms for incentivizing self-interested users) have been addressed in multiagent systems.

We compare our multiagent perspective of crowdsourcing with two other prevalent perspectives: the business perspective and technical perspective [6]. The business perspective mainly considers the business behavior and business principles in crowdsourcing markets; the technical perspective considers the technologies for the development of efficient crowdsourcing systems. Compared with the existing business approaches, the multiagent approach can provide a relatively economical method for investigating crowdsourcing, because the multiagent technology can model and simulate the complex crowd behaviors that are involved. Compared with the existing technical approaches, the multiagent approach represents a more effective and systematic modeling method for investigating human and social behaviors in crowdsourcing.

In summary, the main contribution of this article is a general survey on the state of the art of a comprehensive set of crowdsourcing systems from a multiagent perspective and future research directions for overcoming existing typical technology challenges by using a multiagent approach, which can correlate the research on crowdsourcing and multiagent systems and inspire an interdisciplinary research between the two areas. For crowdsourcing researchers, this article presents a new viewpoint for understanding and investigating crowdsourcing systems; for Multiagent system (MAS) researchers, this article may motivate them to apply multiagent technologies to solve real problems in crowdsourcing systems. Moreover, to the best of our knowledge, this article is the first systematic review of the association between key elements and key processes in crowdsourcing systems. The remainder of this article is organized as follows: In Section 2, we compare our multiagent perspective with other prevalent perspectives for studying crowdsourcing; in Section 3, we discuss the relationship between crowdsourcing systems and MASs; in Section 4, we review the key elements of crowdsourcing systems from a multiagent perspective and identify future research directions by applying a multiagent approach; in Section 5, we review the key processes of crowdsourcing from a multiagent perspective and identify future research directions by applying a multiagent approach; finally, we present the conclusions of this article in Section 6.

2 COMPARISONS WITH OTHER PERSPECTIVES

We first introduce the business and technical perspectives of reviewing crowdsourcing systems and then compare them with our multiagent perspective.

2.1 Business Perspective

The business perspective is adopted by many current studies and surveys on crowdsourcing systems [6]; one of the reasons for this is that many definitions of crowdsourcing consider it to be a new business solution for outsourcing [7]. From the business perspective, crowdsourcing means that tasks can be outsourced by a new business model that can harness the skills, knowledge, and other resources of a crowd of people via an open call [1]. Many business laws and mechanisms have been used to ensure the economic efficiency of crowdsourcing. For example, regardless of the identities of the crowdsourcing companies and requesters, one of their most important objectives is to reduce costs; thus, many related studies have investigated efficient business mechanisms for realizing this objective [3].

The main advantages of the business perspective are as follows: (1) the business perspective can enable crowdsourcing systems to have business agilities that provide efficient and economic tools for utilizing human intelligence for real business applications; (2) the business perspective can lower the IT requirements of crowdsourcing systems for the execution of complex and creative tasks; (3) the business perspective can effectively utilize economic and business forces to shape crowdsourcing systems, which is crucial for the success of crowdsourcing markets; and (4) the business opportunities and potentials can be followed in crowdsourcing markets and, thus, the business value of crowdsourcing can be well exerted.

However, the business perspective has multiple drawbacks: This perspective often lacks rigorous theoretical analyses, technical schemes, and system design for crowdsourcing systems, which may influence the practicality of some crowdsourcing business models. Moreover, it mainly investigates the business mechanisms and potential applications of crowdsourcing; however, sometimes a promising business objective cannot be realized, because existing crowdsourcing platforms are unable to provide feasible technical foundations.

To address these drawbacks, many researchers now also consider the technical aspects of crowdsourcing systems, as discussed next.

2.2 Technical Perspective

The technical perspective focuses on how to supply crowdsourcing services using technologies and mainly emphasizes technical methods, techniques, and frameworks for solving problems in crowdsourcing [6]. Budget-feasible mechanism design [13], task decomposition in workflows [27], and crowdsourcing infrastructure [12] are the typical technical problems that are addressed.

In related studies, the following issues are investigated from the technical perspective: (1) development of a crowdsourcing system based on available information technologies [4]; (2) defining and designing the software components, technical functions, and data objects to be implemented in a crowdsourcing system [12], such as user management, payment mechanisms, quality control, task decomposition, workflow support, and result aggregation; and (3) designing technical mechanisms for ensuring the optimal operation of crowdsourcing systems, for example, in terms of budget feasibility, incentive compatibility, and near-optimal utility achievement [13].

Moreover, there are application-oriented technical perspectives for specific domains, which mainly focus on the application technologies of crowdsourcing in special domains such as data mining [8], software development [14], and mobile sensing [15]. In these related studies, both the crowdsourcing technologies and the domain technologies are considered and combined.

The advantages of the technical perspective are as follows: (1) it has solid theoretical and technical foundations, and, thus, the research results can be highly rigorous, practical, and provable; (2) it can effectively define the components and functions of crowdsourcing systems by considering currently available information technologies, thereby achieving higher technical feasibility; and (3) it can theoretically optimize the user objectives on the technical level.

However, the technical perspective has the following drawbacks: (1) existing related studies that are based on the technical perspective only consider one aspect of technologies and may overlook the systematic and macroscopic crowdsourcing viewpoint; and (2) results that are based on the technical perspective may be inefficient for achieving business and economic objectives; thus, a rougher but simpler crowdsourcing mechanism may sometimes be more welcomed by users than a technologically advanced crowdsourcing mechanism.

2.3 Multiagent Perspective

A multiagent system is a computing system that is composed of a set of agents that perform tasks [66]. The multiagent perspective mainly considers what multiagent approaches can offer to real systems and how multiagent technologies help analyze these systems. This perspective has already been successfully used to model and investigate many autonomous multi-entity systems such as sensor networks and social networks [50, 82].

In this article, the multiagent approach to crowdsourcing systems is a special type of technical perspective that also considers business factors. Crowdsourcing systems are socio-technical systems and the multiagent approach can provide an effective methodology for understanding human behavior and modeling socio-technical systems [16]. Therefore, this article presents a multiagent perspective for surveying the state of the art of crowdsourcing and a multiagent approach to investigating crowdsourcing.

1) *Compared with the existing business approaches*, the multiagent approach can provide a relatively economical method of investigating crowdsourcing, because the multiagent approach can be used to model and investigate crowd behaviors, coordination among workers and requesters, and the effects of incentive mechanisms in crowdsourcing systems. Second, our perspective can better capture complex emergent phenomena from individual worker-level behaviors, which makes it more relevant in crowdsourcing research than the business perspective, because business research only operates on the level of whole systems. However, crowdsourcing systems are real social-technical systems that include many human and social factors, whereas MASs are artificial and predesigned; thus, some models and mechanisms that are applied to MASs may be impractical for application to crowdsourcing systems. Therefore, we should improve the suitability and practical feasibility of multiagent approaches in crowdsourcing.
2) *Compared with the existing technical approaches,* which mainly relies on various concrete technologies, the multiagent approach provides a more effective and systematic modeling method for investigating human and social behaviors in crowdsourcing. In addition, the multiagent approach can make crowdsourcing more cognitive and social, because current multiagent technologies can provide many mature and effective methods for investigating the cognitive and social characteristics of crowdsourcing systems. Moreover, our approach can satisfy the unpredictability requirements of large-scale dynamic crowdsourcing more effectively than the existing technical perspective, because many related multiagent technologies can be used to improve the self-adaptation and self-organization performance of crowdsourcing systems. However, no matter which multiagent approaches are used, they should be implemented by utilizing specific technologies that are highly practical and feasible. Therefore, in practice, we should explore the combination of the multiagent approach with other technologies.

In summary, our multiagent approach is a special type of technical perspective that can also effectively be connected with the business perspective. We can combine them as follows: The business perspective can ensure the macroscopic objectives, the technical perspective can ensure the technical feasibility of crowdsourcing systems, and our multiagent perspective and approach can provide an intermediate method between those two perspectives that can comprehensively consider the social, business, and technical feasibility of crowdsourcing systems.

3 RELATIONSHIP BETWEEN CROWDSOURCING SYSTEMS AND MULTIAGENT SYSTEMS

3.1 Crowdsourcing Systems

Now we present a typical example of performing a task in a crowdsourcing system [4]. At the Amazon Mechanical Turk website, a requester publishes a task such as image annotation or text editing. Then, the requester allocates the micro-task (e.g., image annotation) directly to a crowd of workers, or the requester decomposes the complex task (e.g., text editing) into a workflow of sub-tasks and allocates the decomposed subtasks to workers. The workers execute the allocated task and return the results to the requester. Finally, the requester aggregates the results and pays the workers who provide correct results. In summary, the concept of a crowdsourcing system can be formally defined as follows:

Definition 1. A crowdsourcing system can be described by a tuple $<T, R, O, W, P_1, P_2, P_3>$, where

- T, R, O, and W represent the key elements of a crowdsourcing system: T denotes the set of outsourced tasks, R denotes the set of requesters who present the tasks, O denotes the system platform that organizes and manages the crowdsourcing process, and W denotes the crowd of workers that will perform the outsourced tasks.
- P_1, P_2, and P_3 denote the key processes in the crowdsourcing of tasks, which often include the pre-execution process (P_1), execution process (P_2), and post-execution process (P_3). The pre-execution process runs from the time at which the requester publishes the task to the time at which the assigned workers start to execute the task, which often includes task decomposition and task allocation; the execution process runs from the time at which the tasks are allocated to workers to the time at which the workers complete the tasks; and the post-execution process runs from the time at which the results are presented by the workers to the time at which the requester obtains the final results and the workers receive the associated rewards, which often includes the aggregation and quality control of results and the reward payment to the workers. P_1 and P_3 are implemented by R and O, and P_2 is implemented by W.

3.2 Multiagent Systems

Definition 2. A multiagent system involves multiple agents that can coordinate to perform complex tasks [17]. We define a formal framework for describing a multiagent system, which involves a set of agents and a set of tasks as a tuple $<T, H, M, A, AP_1, AP_2, AP_3>$, where

- T, H, M and A denote the key elements of a multiagent system for performing tasks: T denotes the set of tasks, H denotes the set of hosts who present the tasks or the users who operate the system, M denotes the set of coordination mechanisms in the system, which often includes negotiation, auction, coalition formation, social law and convention, and A denotes the set of agents that will perform the tasks.
- AP_1, AP_2, and AP_3 denote the key processes in performing tasks by the multiagent system, which often include task analysis and allocation, task execution, and task feedback. In the task analysis and allocation process, the user objectives and agent availability are analyzed, the task is decomposed if necessary, and a mapping between tasks and agents is constructed according to the predefined objectives; task execution can be described via the agents' operations when accessing required resources; task feedback includes result handling, reward for agents, and adjustment or reinforcement learning of systems.

Table 1. Comparison Between Crowdsourcing Systems and MASs (Key Elements)

Crowdsourcing systems	Multiagent systems	Associations	Differences and possible inter-disciplinary research
Tasks (T)	Tasks (T)	Both types of tasks involve the following typical characteristics: complexity, dependency, workflow, structures, and reliability; they can be categorized into simple and complex tasks.	The crowdsourcing tasks should be designed for humans, whereas the agent tasks are normally designed for computers. However, many task mechanisms in MASs can be applied in crowdsourcing, such as task structures and task reliability.
Requesters (R)	Hosts (H)	Both types of systems have the following functions for completing tasks: (1) the analyses, decomposition, allocation, and execution of tasks; (2) the rewards or adjustments after tasks; (3) the organization and control of computational resources; and (4) the incentive and optimization mechanisms.	The requesters often undertake many affairs for tasks. In comparison, the people who host the tasks in MASs do not need to do anything, because the system will perform most affairs autonomously. However, the autonomous agent approach can be applied in crowdsourcing to assist requesters in addressing some affairs, such as task decomposition and allocation.
System platform (O)	System mechanism (M)		A rougher but simpler mechanism may be more useful than a strict but complex mechanism in crowdsourcing systems. In comparison, the mechanisms in MASs are more complex and varied. However, the autonomous and adaptive mechanisms in MASs can be applied in crowdsourcing to provide autonomous crowdsourcing service with the capability of autonomous adaptation in highly dynamic social environments.
Workers (W)	Agents (A)	Both workers and agents have the following factors: behavior characteristics, organizations, and truthfulness properties.	Many workers are non-professional and can only perform simple tasks. In comparison, agents have stronger reasoning and decision-making abilities. However, the coordination technologies in MASs can be used to investigate the coordination among workers for performing complex tasks; moreover, the truthfulness mechanisms in MASs can be used to ensure the reliability of workers in crowdsourcing.

3.3 Comparison between Two Types of Systems

Crowdsourcing systems aim at utilizing human abilities to perform computational tasks that are difficult for computers to process [4], and MASs aim at utilizing computer abilities to autonomously perform computational tasks for humans [18]. We perform comparative analyses between the key elements and processes of the two types of systems, as shown in Table 1 and Table 2.

Table 1 and Table 2 show that the key elements and processes of crowdsourcing have close associations with those of MASs, although there are differences between them. Therefore, it is natural to survey crowdsourcing systems from a multiagent perspective. Moreover, although there are differences between MASs and crowdsourcing systems, there are many possible inter-disciplinary research issues between them. Therefore, the multiagent systems can provide an efficient approach and can inspire future research directions that enable crowdsourcing research to overcome the challenges in current crowdsourcing systems.

4 KEY ELEMENTS IN CROWDSOURCING

As stated above, the key elements in crowdsourcing include tasks, requesters, workers, and the system platform. A requester releases a task and its associated reward on the crowdsourcing system platform, and the workers will bid for the task. Then, the requester or the system will allocate the task to workers who will execute the task. Finally, the assigned workers will return the answers to the requester, and the requester will aggregate the answers and reward the workers who present correct answers. The framework for the key elements in crowdsourcing is shown in Figure 1. In this section, we will review these crowdsourcing elements by referring to the related

Table 2. Comparison Between Crowdsourcing Systems and MASs (Key Processes)

Crowdsourcing systems	Multiagent Systems	Associations	Differences and possible inter-disciplinary research
Pre-execution process (P_1)	Task analysis and allocation (AP_1)	Both undertake the following affairs: task analyses, task decomposition, and task allocation.	The pre-execution processes in crowdsourcing systems are often implemented by the requester. In comparison, the pre-execution processes in MASs are implemented by the systems autonomously. However, many methods of task decomposition and allocation in MASs can be applied in crowdsourcing to address complex tasks.
Execution process (P_2)	Task execution (AP_2)	Both utilize the available resources (of workers or agents) to execute the tasks in predefined workflows.	In crowdsourcing systems, the workers execute their tasks independently, and the coordination among their execution results is implemented by the requesters. In comparison, the agents will often coordinate with one another to execute the tasks. However, many crowdsourcing platforms have been beginning to aim at complex tasks and the autonomous coordination mechanisms in MASs can be used for workers to execute tasks.
Post-execution process (P_3)	Task feedback (AP_3)	Both consider the result qualities and make adjustments according to the results.	In crowdsourcing systems, the aggregation and quality control of task execution results are mainly considered. In comparison, the learning and adjustment after task execution are mainly considered in MASs. However, current crowdsourcing environments are often dynamic, which requires the crowdsourcing system to learn and adjust to environments; therefore, related learning and adjustment technologies in MASs can be applied in crowdsourcing.

multiagent concepts and technologies. The classification for the state-of-the-art key elements in crowdsourcing is shown in Figure 2. Next, we will introduce them in detail.

4.1 Tasks

In general, the following properties are often considered in MASs: complexity, structure, dependency, workflow, and reliability. Complexity indicates whether the tasks require higher-level skills

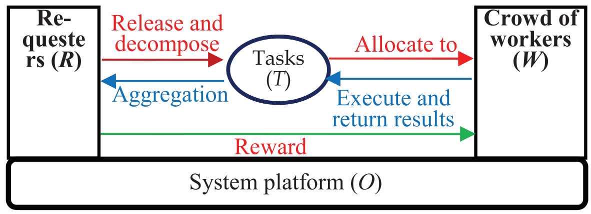

Fig. 1. Key elements in crowdsourcing systems.

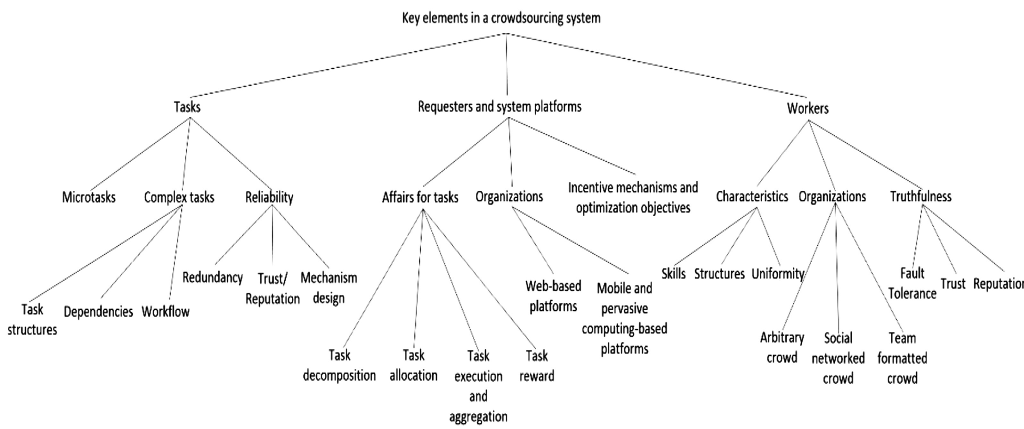

Fig. 2. Classification for the state of the art of key elements in crowdsourcing systems.

and mechanisms [19]; structure, dependency, and workflow define the inherent structural properties of complex tasks [20]; reliability is the probability that the tasks can be executed successfully [21].

In MASs, there are two typical types of tasks: simple tasks and complex tasks [22]. A simple task is often indivisible and can be performed in a prescriptive and straightforward way by a single agent. In comparison, complex tasks often have various inherent structures and can be decomposed into multiple subtasks that are allocated to multiple agents, which requires high-level reasoning and decision-making capacities.

According to the same rule, the outsourced tasks in crowdsourcing systems can also be categorized into two types: micro-tasks and complex tasks [6]. A micro-task is an atomic computational operation and can be completed by an individual worker, whereas a complex task may require multiple skills and cannot be accomplished by an individual worker, such as software development and testing [23], which can be decomposed into a workflow of sub-tasks.

By referring to the typical properties of tasks in MASs, we can summarize the properties of the two types of tasks in crowdsourcing systems, as shown in Figure 3. According to Figure 3, micro-tasks only require consideration of the reliability, whereas complex tasks also require consideration of the structure, dependency, and workflow among sub-tasks.

4.1.1 Micro-Tasks and Complex Tasks.

1) Micro-Tasks

In traditional crowdsourcing markets, the tasks are often simple and micro. Many crowdsourcing platforms, such as Amazon's Mechanical Turk, are oriented to micro-tasks [24]. The micro-tasks,

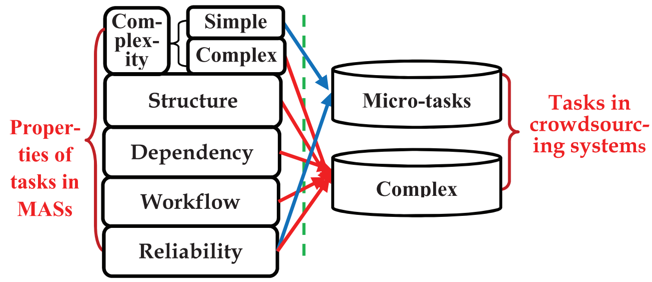

Fig. 3. Relationships between the tasks in MASs and the tasks in crowdsourcing systems.

such as labeling images or rating webpages, can be completed in short amounts of time by non-professional individual workers [4, 23]. The micro-tasks are simple for humans but difficult for computers. For example, finding a building in a picture is very simple for most people but may require substantial computational effort from a computer. Next, we give a formal description of micro-tasks.

Let there be a micro-task, t. Let the set of skills required by t be S_t. Generally, a micro-task can be completed by a single person. $\forall t \Rightarrow \exists w_i \in W \wedge S_{wi} \supseteq S_t$, where W is the crowd of workers and S_{wi} is the set of skills that are possessed by worker w_i.

Because the micro-tasks can be completed by individual workers, the requester only needs to allocate them to workers by considering the workers' skills and task budget constraints. In general, the allocation of micro-tasks can be formalized as follows:

Given a budget b_t that is provided by the requester for a micro-task t, the skills that are necessary for completing t are S_t. Let there be a crowd of workers W. $\forall w_i \in W$, the skills of w_i are S_{wi}, and the reservation wage of w_i is γ_{wi}. Then, the simple task allocation in the crowdsourcing system is defined as task t being assigned to a set of workers W_t, $W_t \subseteq W$, that satisfies the following constraint: Each assigned worker's skills fully satisfy the required skills of task t for independent completion, and the sum of the reservation wages of the workers in W_t does not exceed b_t. Generally, the simple task allocation in a crowdsourcing system can be formalized as follows:

$$\overbrace{\text{Allocates the simple task}(t = <S_t, b_t>)\ to}^{\text{The requester or crowdsourcing system}} \rightarrow \overbrace{\begin{array}{l} W_t = \{w_i | S_{w_i} \supseteq S_t\} \\ s.t. \sum\limits_{\forall w_i \in W_t} \gamma_{w_i} \leq b_t \end{array}}^{\text{Workers}}. \tag{1}$$

In the crowdsourcing of micro-tasks, each task is redundantly allocated to more than one individual worker to improve the accuracy; each assigned worker fully satisfies the required skills of the task and executes the task independently. Finally, the requester will select the correct result from the multiple answers from the redundantly allocated individual workers [4]. Therefore, many existing studies have investigated how to maximize the number of workers who are allocated for a micro-task under a predefined budget constraint or how to achieve a tradeoff between budget and quality in completing a micro-task [25].

A typical application area of crowdsourcing micro-tasks is data mining [8, 25]. For example, mining tasks can benefit from the aggregation of labeling work, which can be easily completed by current crowdsourcing platforms. Classification in data mining is often outsourced through crowdsourcing platforms. A detailed review of studies that are related to the application of crowdsourcing in data mining can be found in Reference [8].

The micro-task-suited crowdsourcing platforms can be extended for complex tasks. For example, CrowdForge [26] addresses the use of micro-task markets to provide scaffolding for more complex tasks that require coordination among many workers, such as writing an article.

2) Complex-Tasks
In real crowdsourcing environments, there are many complex tasks that involve many computational operations and require multiple skills that cannot be possessed by a single worker. Similarly to the methods in MASs, to implement the crowdsourcing of a complex task in a micro-task-oriented crowdsourcing platform, a popular method is to decompose the complex task into a flow of simple sub-tasks. Then, the requester will aggregate the partial results of sub-tasks to obtain the final answer [2, 27]. We now present a formal description of crowdsourcing for complex tasks.

Let there be a complex task t and a given budget b_t for t. First, task t is decomposed into λ micro-subtasks: $t = \{t_m | 1 \leq m \leq \lambda\}$. Then, for each micro-subtask t_m, the requester or the system will assign t_m redundantly to a set of workers such that each worker possesses the required skills of t_m and the total reservation wages of *all* workers who are assigned to *all* micro-subtasks cannot exceed b_t. Let S_i denote the set of skills that are possessed by worker w_i and S_{tm} denote the set of skills that are required by t_m. Such an allocation process can be formalized as follows:

$$\overbrace{\begin{array}{l}\text{Decomposes } t \text{ to a flow of micro-subtasks:}\\ t = \{t_m | 1 \leq m \leq \lambda\}.\\ \forall t_m \in t : \text{allocates } t_m \ to\end{array}}^{\text{The requester or crowdsourcing system}} \rightarrow \overbrace{\begin{array}{l} W_{t_m} = \{w_i | S_{w_i} \supseteq S_{t_m}\}\\ s.t. \sum_{\forall t_m \in t} \sum_{\forall w_i \in W_{t_m}} \gamma_{w_i} \leq b_t \end{array}}^{\text{Workers}}. \quad (2)$$

In both MASs and crowdsourcing systems, the key to performing complex tasks is to decompose tasks into subtasks and coordinate among these subtasks [27, 28]. In general, decomposing a complex task involves the following aspects: the *task structure*, which is the subtask decomposition for the given complex task; *dependencies*, which are the constraints among the subtasks; and the *workflow*, which is the control flow among subtasks [29].

Task structures describe the structural relations among subtasks [30]. There are various typical task structures for complex tasks, such as directed acyclic graphs (DAGs), hierarchical graphs, contract net structures, and arbitrary graphs. The interaction edges in the task structures are composed of the dependency relations among subtasks.

There are many types of *dependency relations* among subtasks, such as time-dependency, which denotes that a subtask must be executed after another subtask; execution-dependency, which denotes that a subtask's execution result is used as the input for the following subtask; and critical-resource-dependency, which denotes that more than one subtask may compete for a critical resource [21]. Alternatively, Ref. [31] categorizes the interdependencies among subtasks into three types: enabling constraints, which denote that a subtask must be executed after another subtask; facilitation constraints, which denote that a subtask improves the quality of another subtask; and hindering constraints, which denote that a subtask decreases the quality of another subtask. Moreover, Tran-Thanh et al. [2] addressed the time and execution dependency relations among subtasks and determined which execution results from one phase should be passed to the next.

A *workflow* utilizes the interdependency relations among multiple sub-tasks to connect them. A workflow structure indicates the temporal relationships among subtasks, which can be represented as a DAG or non-DAG. The workflow control patterns often include the following typical types: sequence, parallelism, choice, and iteration [32]. Dai et al. [33] applied decision-theoretic methods

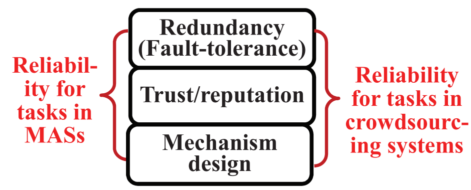

Fig. 4. Summary of the typical types of methods for ensuring reliability of tasks in crowdsourcing systems from a multiagent perspective

of Bayesian network learning and Partially Observable Markov Decision Processes (POMDPs) to optimize workflows in crowdsourcing.

4.1.2 Reliability for Tasks. Reliability for tasks in crowdsourcing systems is the probability that all outsourced tasks can be executed successfully by the crowd. Workers in crowdsourcing systems are often transient and unreliable; moreover, some workers may be malicious and only return generic answers instead of actually executing the tasks to maximize their monetary rewards [4]. Therefore, it is necessary to ensure reliability in crowdsourcing. Next, we will summarize the related studies on reliability for tasks in crowdsourcing from a multiagent perspective.

In general, the measures for ensuring the reliability of tasks in MASs can be categorized into the following three types: redundancy-based approaches, trust/reputation-based approaches, and mechanism design. The redundancy-based approaches achieve reliability via the introduction of redundant copies of tasks so that the system is fault-tolerant if there are mistakes in any single copy of a task [21, 34]. The trust/reputation-based approaches use the concepts of trust and reputation to measure the reliability of agents for tasks, where trust means that one agent is willing to rely on another agent and reputation means that the reliability of an agent is defined by the collective opinions of others [36, 37]. Mechanism design in MASs mainly uses concepts from game theory to design various proper interaction mechanisms for incentivizing selfish agents to perform reliable actions [38].

By referring to the above taxonomy for studies on reliability in MASs, the existing studies in crowdsourcing systems can also be categorized into three types, as shown in Figure 4. The related studies are introduced as follows:

A typical method for reliable crowdsourcing is redundancy of tasks, which is implemented by redundantly assigning each task to more than one worker and combining the answers through various measures, such as majority voting [39]. Therefore, the task allocation objective can be reduced to maximize the number of assigned workers under a budget or minimize the total price under a target overall reliability. Mo et al. [40] investigated how to determine the optimal number of workers for each outsourced task such that the overall reliability is optimized, and they proposed an efficient greedy algorithm that can provide close-to-optimal solutions in practice.

However, the approach of redundancy and majority voting may be infeasible when there are many malicious workers in the crowd [103, 125]. *An alternative method is to use a trust and reputation mechanism.* Yu et al. [41] addressed this problem using trust-aware decision-making approaches for task allocation through crowdsourcing platforms. Venanzi et al. [42] introduced the trust model into a fusion approach for untrustworthy answers that are provided by different workers. Zhang and Schaar [43] proposed reputation-based protocols in crowdsourcing for encouraging workers to perform tasks reliably, which were implemented by integrating reputation mechanisms into a novel game-theoretic model. However, the reputation mechanism may sometimes be infeasible in crowdsourcing systems due to the transient characteristics of workers.

Therefore, in past studies, the reputation-based approach is only utilized in situations in which the workers are experienced and have observed past behavior [41] or where the interactions between requesters and workers are repeated [43].

The third method of ensuring reliability for tasks is based on mechanism design, which aims at designing various mechanisms for incentivizing workers to behave performing tasks reliably [44]. Yang et al. [15] designed an incentive mechanism for motivating smartphone users to reliably participate in mobile phone sensing. They used a Stackelberg game to maximize the utility of the platform and an auction-based incentive mechanism for the user-centric model. Zhao et al. [45] focused on the non-negative monotone submodular value function and investigated online incentive mechanisms for incentivizing workers in mobile crowdsourced sensing to perform truthful actions for reliable task completion.

4.1.3 Summary and Future Research Directions. Next, we summarize some challenges and discuss possible research directions by applying multiagent techniques to tasks in crowdsourcing:

- **Correlated tasks.** In existing studies, most tasks are independent, regardless of whether they are complex or simple. Therefore, the workers who are assigned to a task seldom interact with the workers who are assigned to another task. However, in practice some tasks in crowdsourcing may be correlated and are even constrained by one another [46, 47]. Applying constraint satisfaction and task coordination technologies from the multiagent domain to address the coordination among correlated tasks is a promising approach.
- **Dynamic tasks.** In most current studies, the tasks are static, i.e., the tasks are unchanged from the time at which they are released by the requesters to the time at which they are completed. However, sometimes the tasks may be dynamic during the crowdsourcing process [48]. For example, in mobile sensing environments, the tasks may be adjusted for dynamic targets. Therefore, allowing the workers to adaptively address dynamic tasks is a key problem. To solve this problem, the following two approaches may be attempted: (1) design various adaptive and learning mechanisms for incentivizing the workers to adapt to the dynamic tasks, which can introduce related concepts and methods from the multiagent domain, and (2) equip the workers with assistant software agents to help them address the dynamic tasks.
- **Requester-worker interactive tasks.** Currently, most studies do not consider the interactions between requesters and workers in executing tasks. However, in practice, some tasks may require interactions between them [46, 121]. For example, a dynamic complex task may need to receive real-time directions from the requester. Human-agent interaction has been successfully applied in the execution of participatory tasks in MASs. Therefore, existing technologies of human-agent interaction may provide a promising inspiration for addressing requester-worker interactive tasks in crowdsourcing.

4.2 Requesters and System Platforms

In crowdsourcing systems, the functions of requesters and system platforms may sometimes intersect with each other. For example, in some systems, the task decompositions are implemented by the requesters, whereas in other systems, the task decompositions are implemented by the system platforms. Therefore, in this article, we review the requesters and system platforms together.

Now, we will attempt to review the requesters and system platforms by drawing inspiration from the MAS domain. In a general MAS, the system often involves the following factors in performing a task: (1) the analyses, allocation, and execution of tasks [22, 120]; (2) the rewards and adjustments after completing tasks [54]; (3) the organization and control of agents [55]; and (4) the incentive mechanisms and optimization objectives [22, 121].

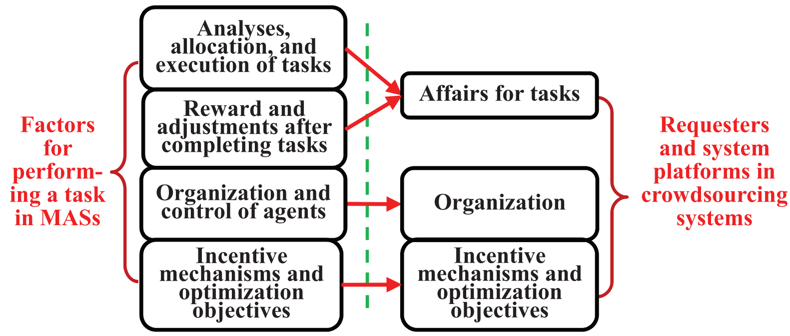

Fig. 5. Review of the requesters and system platforms in crowdsourcing systems from a multiagent perspective.

When a task arrives at a MAS, the system will first analyze the task based on various factors, such as the task objectives, the task structures, and the task's required resources [17]. The system will decompose the task (if the task is too complex to be executed directly by the agents) and allocate the decomposed subtasks to agents according to specified criteria. Then, the allocated agents will access the necessary resources for performing the operations to execute the tasks [19]. After the tasks have been completed, the MAS will make adjustments according to the feedback of the executed tasks. To implement the tasks efficiently, the agents should be organized and controlled efficiently. Thus, the organization and control mechanisms are crucial. Moreover, the organization and control mechanisms have various optimization objectives, such as minimizing the execution time of tasks or minimizing the costs of executing tasks [21].

Therefore, based on the above general factors for performing tasks in MASs, we also review the requesters and system platforms in crowdsourcing by considering the following factors: (1) *affairs that are undertaken by the requesters or system platforms*, such as decomposing the tasks and building the workflows, allocating the tasks to crowds, aggregating and verifying the answers from workers, and distributing the rewards to the workers; (2) *organization of crowdsourcing systems*, which may include the organization of entire systems and the control of the crowdsourcing process; and (3) *incentive mechanisms and optimization objectives of the crowdsourcing systems*, such as monetary and intrinsic incentives for workers and minimizing budgets or maximizing the accuracy of answers. The corresponding relations between the requesters and system platforms in crowdsourcing systems and the factors for performing tasks in MASs are shown in Figure 5.

4.2.1 Affairs for Tasks. As the detailed affairs for performing outsourced tasks that are implemented by the requesters or system platforms will be introduced in Section 5, we will only categorize the crowdsourcing affairs and compare them with the related affairs in MASs in this section. The comparisons are shown in Table 3.

4.2.2 Organization Types of System Platforms. There are various crowdsourcing system platforms, which can generally be categorized into two types: (1) *web-based platforms*, which use Web 2.0 technology and appear as crowdsourcing websites where the requesters post their tasks and the workers register to select tasks to complete for incentives, such as Amazon's Mechanical Turk (MTurk) and Yahoo Answers, and (2) *mobile and pervasive computing-based crowdsourcing platforms*, such as mobile phone sensing systems [15] and pervasive urban crowdsourcing systems [57].

Table 3. Comparisons between Typical Affairs for Tasks in MASs and the Affairs That Are Undertaken by Requesters (or System Platforms) in Crowdsourcing Systems

Typical affairs for tasks in MASs	Typical affairs that are undertaken by crowdsourcing requesters or system platforms	Comparisons
Task analyses: Analyze the user objectives; decompose the task into subtasks (if necessary) and construct the schedules for these tasks [22, 28].	***Task decomposition***: Decompose the complex tasks into a flow of interdependent micro-tasks [2, 23].	The task decomposition methods in both systems are similar. However, in MASs, the task analyses are implemented autonomously by the system; in crowdsourcing systems, the task decomposition is often implemented manually by the requesters.
Task allocation: Assign each task to agents [17, 21].	***Task allocation***: Outsource each task to workers with the proper skills for completing it [49].	In MASs, the task allocation mainly considers the mapping between the tasks' required capacities and the agents' capacities. In crowdsourcing systems, the task allocation considers not only the mapping between the tasks' required skills and the workers' skills but also the budget of the requester and the reservation wages of the workers.
Task execution: The allocated agents utilize resources and operate to execute the task [19, 51].	***Task execution and aggregation***: The allocated workers utilize their skills to complete the tasks and return the answers to the requester; then, the requester aggregates the answers [42, 52].	In MASs, the agents often coordinate with one another to execute the tasks; therefore, coordination and negotiation are often considered in task execution. In crowdsourcing systems, the workers often execute the assigned tasks independently. Thus, there are few studies that investigate the coordination among workers.
Task feedback: After a task is executed, the system will make adjustments according to the results [53, 54].	***Task reward***: Rewards are granted to the workers who provide correct answers [43, 56].	Many incentive mechanisms in MASs and crowdsourcing systems are similar. However, the related studies in crowdsourcing systems often focus on monetary or reputation rewards, whereas the adjustment objects in MASs are relatively varied.

Platforms of the first type, namely, web-based platforms, are often multi-purpose and adopt a centralized control mechanism. For example, Amazon Mechanical Turk is a popular crowdsourcing website where varying crowdsourcing applications can be conducted [4]. In Mechanical Turk, a requester posts his or her task on the website. Then, task decomposition, task allocation, task aggregation, and payments are implemented by the platform and requesters. CrowdFlower [6] is a crowdsourcing system platform that can distribute tasks over multiple crowdsourcing websites, which is also organized in a centralized manner, i.e., the typical affairs are controlled by the system platform or the requesters. In Yahoo Answers, each requester posts her/his question at the website and will select the result from the pool of answers [58].

Platforms of the second type, namely, mobile and pervasive computing-based crowdsourcing platforms, are often designed for a special purpose and may adopt a self-organization control mechanism. Crowdsensing is a typical application of mobile and pervasive computing-based crowdsourcing, in which workers with mobile sensing and computing devices collectively participate to collect data from their contextual environments [114]. In comparison with web-based crowdsourcing, this type of crowdsourcing is more transient and single purpose. As the devices that are owned by different people are heterogeneous, the coordination among people and the fusion of different types of data are very important for this type of crowdsourcing. Although there is a central crowdsourcer in this type of system, the workers should self-organize for some desirable properties. Therefore, many related studies have investigated incentive mechanisms for motivating the collaboration of workers (such as smartphone users) in performing tasks [59].

4.2.3 Incentive Mechanisms and Optimization Objectives. Incentive mechanisms are necessary for attracting people to participate in crowdsourcing [60]; moreover, a proper incentive mechanism can motivate workers to contribute truthful answers. Yuen et al. [4] categorized the typical incentives in crowdsourcing systems as monetary incentives, which can provide the workers with money, and intrinsic incentives, such as reputation, attention, happiness, and self-satisfaction. Moreover, Pan and Blevis [61] categorized the intrinsic incentives in crowdsourcing systems into social incentives, such as respect from others, social status, and social connection; and personal incentives, such as fun, personal interest, and self-value.

For the various incentives, many related studies have investigated efficient incentive mechanisms for managing the behaviors of workers to achieve various optimization objectives, which are described as follows.

In web-based crowdsourcing systems, various algorithms and game-theoretical techniques have been used to achieve the target reliability or optimal budget. Karger et al. [113] used low-rank matrix approximation to develop an algorithm for constructing proper task assignment schemes and extracting the correct result from workers' answers, which aimed at achieving satisfactory reliability at a minimum total cost. The approach that was presented by Karger et al. significantly outperforms the previous majority voting method, which approaches the ideal situation in which the reliabilities of all workers are known. Moreover, Tran-Thanh et al. [25] proposed a budget allocation algorithm that is based on agents for distributing a specified budget among different tasks to achieve low estimation error, which can significantly outperform the work of Karger et al. Zhang and Schaar [43] integrated reputation mechanisms into a repeated game-based game-theoretic model to make the crowdsourcing website operate close to Pareto efficiency, which can incentivize selfish workers to work hard at performing tasks.

In mobile and pervasive computing-based crowdsourcing systems, some incentive mechanisms that are based on auction and game theory have been presented in past studies. Jaimes et al. [63] presented an incentive mechanism that is based on recurrent reverse auction and designed a greedy algorithm that can allocate a specified fixed budget among workers according to their locations.

Zhao et al. [45] designed online incentive mechanisms for achieving the objectives of individual rationality and computational efficiency for the zero arrival-departure interval case in a realistic mobile crowdsourcing sensing scenario in which workers arrive online in a random order. Yang et al. [15] used an auction-based incentive mechanism for the user-centric model and a Stackelberg game to maximize the utility of the platform, which can motivate smartphone users to participate reliably in mobile phone sensing.

4.2.4 Summary and Future Research Directions. Next, we summarize various challenges and discuss possible research directions in which multiagent techniques are applied to requesters and crowdsourcing system platforms:

- **Autonomous crowdsourcing platforms that can assist requesters in conducting crowdsourcing affairs.** On most crowdsourcing platforms, requesters may need to undertake many affairs, such as task decomposition, task allocation, answer aggregation, and reward distribution [23, 39]. Therefore, some requesters may abandon the use of crowdsourcing platforms, because they cannot bear such a heavy burden by themselves. In the future, autonomous agent technologies can be used by crowdsourcing platforms to assist requesters with many affairs. Then, the requesters can be exempt from a heavy workload when they use the crowdsourcing platforms.
- **Collaboration among multiple crowdsourcing platforms.** Although there are studies on task distribution over multiple platforms, such as CrowdFlower [6], few systematic studies have considered collaboration among multiple crowdsourcing platforms. In the future, collaboration among multiple crowdsourcing platforms with various functions and areas of expertise should be investigated to provide crowdsourcing services for hybrid requesters. There are many studies on inter-platform and inter-organizational collaboration in the MAS domain [64], which can provide basic frameworks for modeling collaboration among multiple crowdsourcing platforms. Moreover, the inter-organizational workflows in the MAS domain [64] can provide a basic theoretical foundation for the distribution and collaboration of complex tasks among multiple crowdsourcing platforms.
- **Component-based adaptive and scalable crowdsourcing platforms.** Most crowdsourcing platforms cannot adapt to changes in environments. Many crowdsourcing factors may change over time, and the scales of requesters, tasks, and workers may vary significantly [48]. Thus, adaptive and scalable platforms are needed. The component technologies and dynamic organizations in MASs can achieve adaptability, robustness, and scalability [65], which can be applied to build adaptive and scalable crowdsourcing systems. Moreover, the learning technologies in MASs can be applied to the adaptive mechanisms of crowdsourcing systems.

4.3 Workers

In MASs, the following aspects of agents are often addressed: characteristics, organization, and truthfulness. Various characteristics are related to each agent, such as their capacities, behavioral strategies, and uniformity [66]. Generally, there are three typical forms of agent organizations: arbitrary organizations [67], network structures [19], and coalitions [68]. Moreover, real MASs may be unreliable due to the heterogeneity and openness of the systems [19]. Therefore, the truthfulness of agents is often investigated [69], through factors such as fault-tolerance, trust, and reputation.

Based on the general factors of agents that are listed above, we also review the workers in crowdsourcing in terms of the following factors: (1) *characteristics of workers*, such as their skills (non-professional vs. professional workers), strategies (cooperative workers vs. non-cooperative

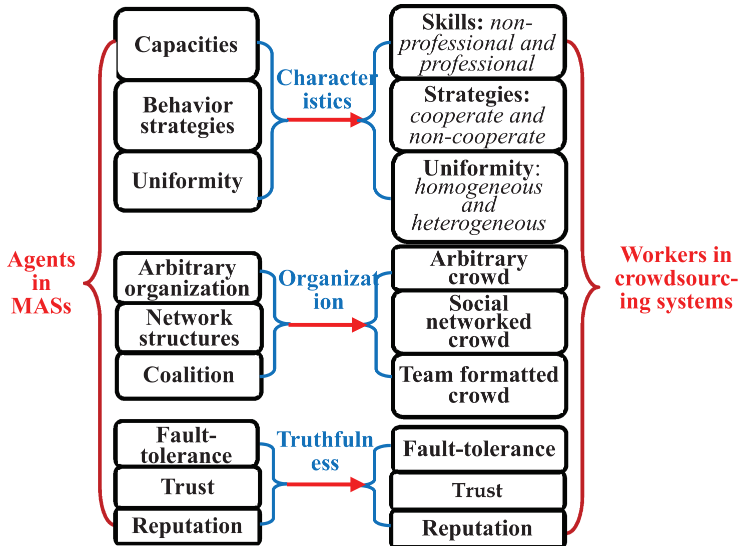

Fig. 6. Review of the workers in crowdsourcing systems from a multiagent perspective.

workers), and uniformity (homogeneous workers vs. heterogeneous workers); (2) *organizations of crowds of workers*, which include arbitrary crowds, socially networked crowds, and team-formatted crowds; and (3) *truthfulness of workers*, which often includes the fault-tolerance of workers, trust among workers, and reputations of workers. The corresponding relations between the workers in crowdsourcing systems and the agents in MASs are shown in Figure 6.

4.3.1 Characteristics of Workers. By referring to the main characteristics of agents in MASs, we review the following characteristics of workers in crowdsourcing systems: skills, strategies, and uniformity of crowds.

1) Skills
The skills of a worker are his/her capacities for completing tasks. Generally, the workers who are recruited by the requesters can be categorized into two types: non-professional and professional workers. Typically, simple micro-tasks or decomposed subtasks can be completed in short amounts of time by non-professional workers [8], such as annotating images or participating in surveys. The related studies seldom conducted systematic research on non-professional workers.

Currently, there are a growing number of real crowdsourcing applications in which tasks require significant efforts by professional workers. The crowdsourcing of these tasks is called expert crowdsourcing [23, 115]. Crowdsourcing to professional workers is often oriented toward complex task applications such as the development of software, writing scientific articles, and building websites. Bozzon et al. [70] presented a method for finding the most knowledgeable people in social networks to address tasks. Tran-Thanh et al. [23] investigated the problem of expert crowdsourcing based on a novel multi-armed bandit model. Kulkarni et al. [71] presented a system, which is called Wish, for identifying and recruiting experts from an online crowd to accomplish complex creative tasks.

Moreover, the combination of professional and non-professional workers may improve the performance. For example, Baba et al. [72] presented a method for recruiting non-professional workers and professional workers to detect improper tasks in crowdsourcing marketplaces, which can improve the detection performance. There are studies on cooperation among experts in performing tasks, such as on finding a team of experts in social networks to not only satisfy the skill requirements of the task but also cooperate effectively [73].

2) Strategies
The behavioral strategies of workers can be categorized into two types: cooperative and non-cooperative. Cooperative workers work toward satisfying the same goal, whereas non-cooperative workers are often self-motivated and try to maximize their own benefits.

Some crowdsourcing markets that are oriented toward workers in social networks assume that the workers are cooperative. For example, team formation, which aims at finding a group of workers that match the skill requirements of outsourced tasks, often requires the workers to be cooperative. Lappas et al. [73] and Majumder et al. [74] presented efficient approximate algorithms with provable guarantees for finding near-optimal teams, which can solve the NP-hard problem of traditional team formation of cooperative workers in social networks.

However, many workers in crowdsourcing systems may be non-cooperative. To make the non-cooperative workers perform the tasks successfully, some incentive mechanisms have been investigated in related studies. For example, Jain et al. [75] addressed the situation in which there are non-cooperative strategic consumers with unknown response characteristics and proposed a multi-armed bandit mechanism for crowdsourcing demand response. Moreover, in the mobile and pervasive computing-based crowdsourcing systems, the workers are often non-cooperative. Thus, many studies on these systems have investigated incentive mechanisms for encouraging non-cooperative workers to participate in the tasks. For example, Zhao et al. [45] presented an online incentive mechanism for encouraging non-cooperative smartphone users to complete outsourced tasks truthfully within budget constraints. More related studies on the incentive mechanisms for non-cooperative workers can be found in Section 4.2.3.

3) Uniformity
In MASs, agents may be homogeneous or heterogeneous [76]: Homogeneous agents are often attributed to the same organization and have the same characteristics, whereas heterogeneous agents have different characteristics. The workers in crowdsourcing systems may also be homogeneous or heterogeneous. For example, the workers who are registered at an insect-fan website may be homogeneous, whereas the workers who are registered at a general crowdsourcing website may be heterogeneous [4]. Typically, homogeneous workers can only solve simple and homogeneous tasks. Heterogeneous or complex tasks are outsourced to a crowd of heterogeneous workers [77]. In real environments, there are few cases in which workers are homogeneous. Many workers are heterogeneous with different skills or reservation wages. For example, the study in Reference [23] addressed heterogeneous workers with different costs and answer qualities. Bernstein et al. [78] introduced the Find-Fix-Verify crowd programming pattern and investigated the allocation of heterogeneous workers within a multiple-phase outsourced task.

4.3.2 Organizations of a Crowd of Workers. Agents are often organized into various forms, which can improve their performance; an effective organization can maximize the overall utilities of agents [79]. Crowds of workers should also be organized into specific forms. Generally, the following organizational forms are often observed in real-world crowdsourcing systems: arbitrary crowds, socially networked crowds, and team-formatted crowds.

Initially, in earlier crowdsourcing platforms, any worker could apply to undertake the posted tasks. Thus, such workers were random and transient and had no predefined organizations [80]. In situations in which the workers are arbitrary, the workers only need to report their skills and reservation wages when they want to bid on the posted tasks.

With the development of social networks, people are often structured and organized through social networks and can solve complex tasks [81, 82, 83, 116]. Chamberlain [81] proposed a definition for groupsourcing: a group of people who are connected through a social network is recruited to complete a task and the task requester takes part in the organization of the group.

Team formation is a new organization method for performing complex tasks, in which workers with different skills form a team to complete tasks collaboratively [84]. The previous works on this subject can be categorized into centralized and self-organized approaches according to the control mechanisms.

In the centralized approaches, the team formation of workers is fully controlled by the requester or the crowdsourcing system. In the representative mechanisms that were presented by Liu et al. [84], the requester recruits workers according to their skills and bid prices to form a valid team for completing the task. Then, the requester can decide each individual worker's payment. Moreover, the workers are sometimes organized in a social network and the communication between any two workers may incur certain communication costs. Then, the requester's objective is to find a team of workers that can minimize the communication costs among workers in the team while satisfying all of the skills that are required by the tasks, which is NP-hard. To solve this NP-hard problem, Kargar et al. [85] proposed a bounded approximation algorithm with a proved approximation ratio, which was scalable and effective on real datasets.

In the self-organization approaches, individual workers must self-form a team through their local visibility in the crowd. Rokicki et al. [86] explored a self-organization method for team formation in which each worker initially forms a one-mate team and workers can select any teams in which to participate; two teams can be merged to a larger team and teams can compete against one another in bidding for tasks. Singla et al. [87] formalized the self-organized team formation problem as a function maximization problem with local knowledge of the crowd in a decentralized manner. Osipov and Sukthankar [14] designed a prototype, which is called AmalgaCloud, for allowing workers to self-initiate team formation proposals and choose among alternative proposals.

4.3.3 Truthfulness of Workers. Because crowdsourcing systems are open, workers are often transient and unreliable. Moreover, there may be malicious workers who only return generic answers instead of actually executing the tasks to maximize their monetary rewards [4]. Therefore, the truthfulness of workers has been widely studied [88]. The following aspects were investigated to ensure the truthfulness of workers: fault-tolerance [62], trust [89], and reputation [41]. The related studies on these aspects attempt to ensure the reliability of tasks. Therefore, the review of the truthfulness of workers is similar to the review of the reliability of tasks in Section 4.1.2. We skip discussing these similar topics again here due to space limitations.

4.3.4 Summary and Future Research Directions. Next, we summarize various challenges and discuss possible research directions that involve applying multiagent techniques to workers:

- **Collective, dynamic, and self-organized coordination among workers.** In most existing crowdsourcing systems, workers often execute tasks independently and the requester is responsible for coordinating the results among workers. Many workers may need to coordinate to execute complex tasks and adapt to dynamic environments [47, 90]. In the future, we can apply the coordination mechanisms and collective decision-making mechanisms of MASs to investigate the coordination of workers.
- **Modeling the thinking, reasoning, creativity, and other intelligent behaviors of workers.** In existing studies, the workers are often assumed to execute the tasks in a straightforward manner. However, when workers execute complex or creative tasks, thinking, reasoning, and other intelligent behaviors are required [26, 91, 92]. Therefore, modeling

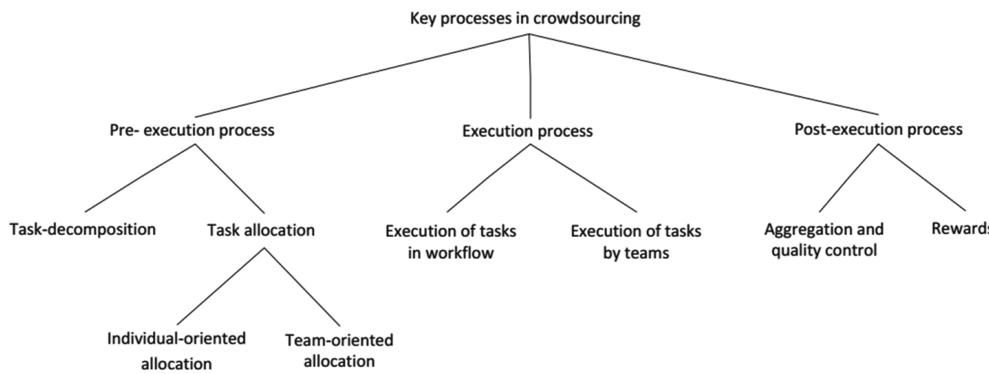

Fig. 7. Classification for the state of the art of key processes in crowdsourcing.

these intelligent behaviors of workers is crucial for the design of efficient crowdsourcing mechanisms and crowdsourcing platforms. In the future, many related models and mechanisms of intelligent agents can be introduced to solve this problem.

- **Dynamic strategies of workers.** In current related studies, the strategies of workers are always assumed to be fixed. However, workers may sometimes change their strategies dynamically, e.g., a worker may select a cooperative or non-cooperative strategy according to the current situation [15]. In the future, we can introduce the dynamic mechanism design of MASs to address this problem.

5 KEY PROCESSES IN CROWDSOURCING

Crowdsourcing systems aim at performing tasks that are trivial for humans but difficult for computers. We now discuss the processes of crowdsourcing tasks and divide crowdsourcing into three key processes for performing tasks: the pre-execution process, the execution process, and the post-execution process. The pre-execution process and the post-execution process are implemented by crowdsourcing platforms and requesters, and the execution process is implemented by workers. The classification for the state of the art of key processes in crowdsourcing is shown in Figure 7. Next, we will introduce them in detail.

5.1 Pre-Execution Process

When a MAS wants to perform a task, it will first analyze the user desires and agent availability and decompose the task if necessary [28]. Then, the system will perform task allocation by mapping between tasks and agents to satisfy the predefined objectives. Similarly, when a task is posted by a requester on the crowdsourcing platform, the task may be decomposed into sub-tasks that are structured in a workflow if necessary. Then, the system or requester will assign the tasks (or de-composed sub-tasks) to workers who have the necessary skills to complete the tasks and who can satisfy the specified constraints and objectives. Therefore, the two processes in MASs and crowdsourcing systems can be correlated with each other. We now review two important aspects of the pre-execution process: task decomposition and task allocation.

5.1.1 Task Decomposition. Given a task to perform, a MAS first determines whether the task can be decomposed into subtasks that can be performed concurrently and takes advantage of the teamwork performance among agents to maximize the global objective [22, 93]. Similarly, to implement the crowdsourcing of complex tasks, a popular approach is to decompose each task into a flow of simple subtasks [2].

Fig. 8. Taxonomy of task allocation in crowdsourcing systems.

Jiang and Matsubara [27] addressed the task decomposition problem in crowdsourcing and defined two types of task decomposition: vertical task decomposition for dependent subtasks and horizontal task decomposition for independent subtasks.

A representative workflow for decomposing tasks was presented by Bernstein et al. [78], in which each complex task can be split into Find-Fix-Verify phases. The problem is determined in the Find phase and fixed in the Fix phase and the quality of the results is controlled in the Verify phase. Moreover, Ambati et al. [90] decomposed the complex task of translation into a workflow that includes lexical translation, assistive translation, and monolingual synthesis.

A key problem in task decomposition in crowdsourcing is determining the number of subtasks in each phase and the amount of payment for each sub-task. For example, Tran-Thanh et al. [2] proposed the first crowdsourcing algorithm, which is called BudgetFix, which can guarantee result quality by determining the optimal number and prices of decomposed micro-tasks under given budget constraints.

In addition to the requesters, workers can also aid in performing task decomposition. For example, Kulkarni et al. [46] presented Turkomatic, which is a tool for recruiting workers to aid in the decomposition of tasks. With the tool, the requesters can observe and adjust the task decomposition that is designed by workers in real time. This collaborative approach outperforms traditional task decomposition, which is implemented only by requesters.

Although most related studies have focused on the resulting quality of task decomposition, the throughput of sub-tasks is another factor to be considered. For example, Sautter and Böhm [94] presented scalable crowdsourcing mechanisms for the decomposition of complex tasks with the objective of high throughput.

5.1.2 Task Allocation. In MASs, task allocation is the key problem of finding a task-to-agent mapping that optimizes specified global objectives, such as minimizing execution time, maximizing social utility, and maximizing throughput [21, 22]. For simple tasks, the system can allocate tasks directly to individual agents. For complex tasks, there are two methods for performing task allocation [117]: (1) the tasks are decomposed into subtasks and the subtasks are allocated to individual agents [93], or (2) the tasks are directly allocated to a team of agents and the agents coordinate to execute the tasks [95].

By drawing inspiration from the above typical task allocation methods in the MAS domain, we also review the task allocation for simple tasks and complex tasks in crowdsourcing systems. For the crowdsourcing of simple tasks, the tasks are directly allocated to individual workers [25]; for the crowdsourcing of complex tasks, the tasks may either be decomposed into micro-subtasks to be allocated to individual workers [2] or be directly allocated to a team of workers [84]. Therefore, we categorize the related studies into two types: individual-oriented allocation and team-oriented allocation, which are shown in Figure 8.

1) Individual-Oriented Allocation

Most previous studies have adopted the individual-oriented crowdsourcing method, which means that each assigned worker can complete the allocated task individually and independently. In individual-oriented crowdsourcing, the requester will attempt to maximize the utility from his/her released task within the given budget and the workers will attempt to maximize their own individual utilities by deciding which tasks to perform at what price [13]. Therefore, a tradeoff between the requester and workers is needed.

As stated beforehand, there are two typical types of tasks in crowdsourcing systems: micro-tasks and complex tasks [6]. Thus, we will review the individual-oriented crowdsourcing approaches for these two types of tasks.

1.1) Individual-oriented allocation of micro-tasks

Micro-tasks are atomic computation operations and can be completed in short amounts of time by non-professional individual workers. In the crowdsourcing of micro-tasks, each task is redundantly allocated to more than one worker to improve the accuracy, each worker's skills fully satisfy the required skills of the task, and each worker executes the task independently. Finally, the requester will select the correct result from the multiple answers from the redundantly allocated individual workers [4]. Next, we will introduce some representative studies.

Tran-Thanh et al. [25] presented an agent-based algorithm, which is named CrowdBudget, for redundant task allocation in which the estimation error is at most $\max\{0, K/2\text{-}O(\sqrt{B})\}$, where B denotes the budget and K is the number of tasks. Goel et al. [35] designed an incentive-compatible mechanism, which is named TM-UNIFORM, for task allocation with the following two properties: truthfulness and budget feasibility.

Ho et al. [49] extended the online primal-dual technique that is used in the online adword problem and presented a two-phase exploration-exploitation assignment algorithm for solving the online task allocation problem. He et al. [96] addressed the allocation of crowdsensing tasks that are associated with specific locations and presented a local ratio-based algorithm for optimal allocation of location-dependent tasks.

Another representative approach for task allocation is to investigate the problem of minimizing the number of task assignments to achieve a target reliability. Karger et al. [39] presented a non-adaptive task allocation algorithm that was inspired by low-rank matrix approximation and belief propagation and can achieve optimal reliability by comparing to an ideal situation in which the reliability of every worker can be known.

Moreover, multiagent technology can be used to automate task allocation in crowdsourcing. For example, Chen et al. [97] used multiagent planning and a stochastic recommendation approach to realize automatic task allocation in mobile crowdsourcing, which is implemented based on workers' desired time budgets and historical trajectories.

1.2) Individual-oriented allocation of complex tasks

Complex tasks involve many computational operations and require multiple skills, and they cannot be completed directly by individual workers. To accomplish a complex task using a micro-task-oriented crowdsourcing platform, a popular method is to decompose the complex task into a flow of simple subtasks and allocate each sub-task to individual workers [2].

A representative work is that of Tran-Thanh et al. [2], in which a crowdsourcing algorithm, which is named BudgetFix, was proposed. BudgetFix can dynamically allocate its budget to each micro-task and allocate a proper number of micro-tasks at each phase of a workflow. Let B be the budget limit. The algorithm can provably achieve an accuracy probability of $1\text{-}e^{-O(B)}$.

Another representative work is that of Kittur et al. [26], in which a general-purpose framework, which is named CrowdForge, was developed using a web-based prototype in a software toolkit. The framework can accomplish complex and interdependent tasks based on micro-task platforms and provide typical functions for task allocation, such as partitioning, mapping, and reduction.

2) Team-Oriented Allocation of Complex Tasks

As stated above, many complex tasks cannot be completed directly by individual workers. Although many existing approaches decompose complex tasks and allocate the decomposed subtasks to individual workers, this may produce heavy decomposition loads. Therefore, there has been an emergence of studies that attempt to directly allocate complex tasks to a team of workers, which is called team-oriented crowdsourcing. In related work, team formation has been intensively investigated. Team formation is a new method for crowdsourcing complex tasks, in which individuals with different skills form a team for completing tasks collaboratively [84].

There are similar concepts in the domain of MASs, such as coalition formation [68] and team formation [98]. A coalition or team can be formed through a central controller or distributed negotiation among agents. Team formation in crowdsourcing systems can also be considered a special form of team formation in MASs but with the objective of completing the outsourced complex tasks efficiently to maximize specific objectives.

As stated in Section 4.3.2, a team can be formed either by centralized approaches, where the team formation of workers is fully controlled by the requester, or by self-organized approaches, where individual workers need to self-form a team through their own knowledge and decisions. After a team has been formed, the tasks can be allocated to the team for execution.

5.2 Execution Process

The task execution step represents the actual execution of the task by the assigned workers [99]. In fact, there are relatively fewer studies focusing on the execution process in crowdsourcing systems. Especially, there are almost no related studies on the execution process of simple tasks, because simple tasks can be completed by utilizing workers' skills in a straightforward manner. We now only introduce the execution process of complex tasks.

5.2.1 Execution of Tasks in Workflows. After a complex task is decomposed into subtasks structured in workflows, the allocated workers will execute the subtasks in each stage of the workflows. Related studies have often considered the concrete processes of stages and the constraints among stages of the workflow. A representative work is the study of Bernstein et al. [78], which presented a word processing interface based on Mechanical Turk to perform complex tasks, such as proofreading and editing documents. The execution process can be divided into three stages: Find, Fix, and Verify. The patches of the requester's task will be identified by workers in the Find stage. Then, the patch identified by the Find stage will be revised by other workers in the Fix stage. Finally, the best answer will be voted on, and the quality control will be performed in the Verify stage. Therefore, in these stages of workflows, the results of the previous stage will be forwarded to the next stage. Another representative work is that Ambati et al. [90] presented a pipeline model for the execution of crowdsourcing translation, where the output from previous stages can be polished in the subsequent stages. In their model, the execution process also includes three stages: The workers perform translations at the word/phrase level in the first stage; the workers collect complete sentence-level translations in the second stage; and, finally, the workers extract a new translation from the multiple translations produced in the previous stage.

Moreover, to manage the workflows efficiently, Kittur et al. [26] designed a system, CrowdWeaver, which can visually manage the task execution process to control the execution of complex crowd workflows, such as the tracking and notification of task progress, real-time modification of the workflow, and the flow of data between tasks. In Reference [46], Kulkarni et al. proposed a novel task execution method where the requesters can observe and decide whether intervene to control quality in the execution process. If the quality is not satisfactory, then the requesters can even modify specific components of workflows during the task execution process.

In the task execution process, decision-theoretic approaches are used to dynamically adjust the settings of workflows and interaction among multiple workflows can improve the quality of task execution [100]. For example, Dai et al. [33] designed an approach that uses POMDP and Bayesian learning theory to perform dynamic control and implement switching between alternative workflows for executing tasks, which can be provably demonstrated useful on Amazon Mechanical Turk. Moreover, Lin et al. [101] presented a POMDP-based controller to make dynamic switching among multiple workflows to achieve higher quality results than a single workflow in task execution processes.

5.2.2 Execution of Tasks by Teams. As mentioned in Section 5.1, after a team is formatted for a task, individual workers in the team will complete the task collaboratively. We will now introduce some studies on the collaboration between teams for the execution of tasks.

Park et al. [102] addressed the industrial product design problem in crowdsourcing systems. As industrial product design problems are often too complex to execute, they presented the Crowd vs. Crowd (CvC) method, which formed multiple design teams to compete with each other to produce competent design results. In each team, there is a coordinator who controls the communication between the team members and monitors the design result.

In Reference [86], Rokicki et al. presented a series of dynamic team competition strategies for executing outsourced tasks: self-organizing teams, including balanced teams, and the combination of individual and team strategies. In the execution process, the workers can decide which team they want to join by considering real situation.

5.3 Post-Execution Process

After the tasks are executed by workers, some measures will be made to finalize the crowdsourcing process, which is called the post-execution process. In general, the post-execution process includes the following typical aspects: aggregation and quality control of results and distribution of the rewards to the workers. In the current related work, the post-execution process may be implemented either manually by the requesters or automatically by the crowdsourcing systems.

5.3.1 Aggregation and Quality Control of Results. As stated beforehand, the tasks may be redundantly outsourced to more than one worker to achieve accuracy and reliability, thus several redundant solutions may be reported for the same task. Therefore, the aggregation of results is necessary for achieving a final solution [103]. Moreover, some workers may have poor talent or be malicious, and thus the quality control of the results is crucial.

Majority voting is often used to implement aggregation to ensure result quality, where the requester simply adopts the answer that is agreed on by the majority of workers [62]. However, such simple majority voting is only effective in the situations where the agreement degree of workers is high. To address the situation where the answers of different workers vary greatly, Yue et al. [104] presented a weighted majority strategy wherein each worker is associated with a weight based on his/her accuracy on the tasks. Moreover, the simple majority voting method often assumes that all workers and all tasks are homogeneous. To solve this problem, Georgescu and Zhu [105] presented a novel aggregation method wherein each worker is associated with a proportional weight regarding with the worker's expertise.

Matsui et al. [106] introduced the abilities of workers into the item ordering questions, and they extended a traditional distance-based order model to a probabilistic generative model of crowd answers. Sometimes the workers are anonymous; thus, Hui et al. [107] tested the effects of anonymity of workers on improving the quality of online answers. They found that anonymous workers can provide more specific criticism and specific praise.

5.3.2 Rewards. The most typical rewards in related work are monetary reward and reputation [80]. The monetary rewards can influence the workers' willingness to perform tasks and affect both the quality and quantity of workers' work [108]. Generally, the monetary reward to a worker is related to the worker's participation levels and answer qualities in performing the outsourced tasks [109]. The workers with good performance may be rewarded by the requester, but the workers with poor performance may be not rewarded by the requester [110]. In the redundant allocation of simple tasks, all workers who provide the correct answers will be paid. With the complex tasks, often the workers providing the best answer, i.e., the solution chosen as the winner, is paid [91, 109]. Moreover, some answers are associated with certain contexts and the difficulties of providing answers with different contexts are different, thus the contexts of answers may influence the actual rewards for the workers. For example, Biswas et al. [60] addressed the reward mechanism in the participatory sensing for smart cities, where the rewards are distributed according to the spatiotemporal contexts of sensing reports. In their reward mechanism, the workers reporting traffic congestion information during peak hours from a business district may be rewarded more than the workers reporting traffic information from a sparse residential area.

A worker's reputation is related to the worker's experience for completing outsourced tasks, which can reflect the worker's truthful ability and can influence the worker's probability to be recruited in the future. Daltayanni et al. [111] proposed a novel reputation mechanism based on Bayesian update method that can consider the requester's implicit feedback information and solve the problems of previous reputation approaches in which the reputation values were usually undependable. Moreover, to address the situation where the reputation score may not truly reflect the worker's future performance due to the highly heterogeneous task categories, Kokkodis and Ipeirotis [112] considered the task categories and used prior and category-specific feedback to construct inter-category reputation for a worker, which can improve the accuracy compared to the previous reputation approaches.

5.4 Summary and Future Research Directions

Next, we summarize various challenges and discuss possible research directions that involve applying multiagent techniques to the crowdsourcing process:

- **Concurrent processes of multi-tasks.** In most of the current related studies, only single-task processes are investigated. However, in practice, there are interrelated tasks running concurrently [46, 101]. Concurrent processes of multi-tasks may bring new challenges to the scheduling and coordination of tasks and the decision-making of crowdsourcing systems for the management of large-scale crowdsourcing processes. Currently, there are many mature multiagent technologies for the dynamic scheduling of concurrent tasks, multiagent decision-making models for large-scale process management, and related mechanisms of multiagent-based pipeline scheduling optimization [20, 30, 38]. Those scheduling, decision-making, and coordination approaches for concurrent tasks in the multiagent domain can be introduced into crowdsourcing systems in the future.
- **Competition among different crowdsourcing processes.** When many interrelated tasks are executed concurrently by the same crowdsourcing system, the different processes may compete for various critical resources of the system and even for critical workers [46]. The resource allocation and collaborative planning technologies in MASs may provide useful methods and research directions. For example, there are market-based incentive mechanisms that allow agents to distribute resources for maximizing social welfare, which can be introduced into the coordination of requesters for using the crowdsourcing resources. The

negotiation-based schemes that are used to coordinate the planning between workflows in MASs can be applied to schedule concurrent crowdsourcing processes.

- **Large-scale, dynamic, and unpredictable crowdsourcing processes.** Some large-scale and dynamic crowdsourcing markets may lead to crowdsourcing processes that are highly dynamic and unpredictable [33, 101]. Thus, it is difficult to give complete *a priori* specifications for all affairs in crowdsourcing processes. Moreover, unanticipated events may occur, e.g., workers may quit executing assigned tasks or the requester may abandon the tasks suddenly. In MASs, there are many related technologies for addressing dynamic business process management using a collection of autonomous agents [118]. In the future, the dynamic process management technologies in MASs can be used to explore the dynamic management of crowdsourcing processes.

6 CONCLUSIONS

There have been significant research results on crowdsourcing in recent years. Most existing surveys on crowdsourcing only conducted a very preliminary review on a single aspect of crowdsourcing systems or on the application of crowdsourcing to a specific application domain. To present a more general and macroscopic survey, this article uses a multiagent perspective to review the state of art of a comprehensive set of crowdsourcing systems, including (1) the key elements of crowdsourcing systems, which include tasks, requesters, system platforms, and workers, and (2) the key processes of crowdsourcing tasks, which include the pre-execution process, the execution process, and the post-execution process. Moreover, this article uses a multiagent approach to identify future research directions that will enable crowdsourcing research to overcome typical challenges in key elements and processes in crowdsourcing.

In the future, several important research issues need to be addressed to truly and effectively utilize the multiagent approach in crowdsourcing. First, additional social mechanisms of the multiagent domain can be used to model crowdsourcing systems and design business mechanisms, such as social choice and social welfare, social law and convention, and artificial society systems. Second, additional coordination mechanisms of the multiagent domain can be used to design the control mechanisms of crowdsourcing systems, such as game theory and mechanism design. Third, additional learning and adaptation mechanisms in the multiagent domain can be used to address the dynamics and evolution of crowdsourcing systems.

REFERENCES

[1] J. Howe. 2006. The Rise of Crowdsourcing. *Wired Magazine*, 14, 6 (2006), 1–4.
[2] Long Tran-Thanh, Trung Dong Huynh, Avi Rosenfeld, Sarvapali Ramchurn, and Nicholas R. Jennings. 2014. BudgetFix: Budget limited crowdsourcing for interdependent task allocation with quality guarantees. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'14)*. 477–484.
[3] T. Kohler and M. Nickel. 2017. Crowdsourcing business models that last. *J. Bus. Strategy* 38, 2 (2017), 25–32.
[4] M. C. Yuen, I. King, and K. S. Leung. 2011. A survey of crowdsourcing systems. In *Proceedings of the IEEE 3rd International Conference on Social Computing (SocialCom'11)*. 766–773.
[5] Q. Li, A. Vempaty, L. R. Varshney, and P. K. Varshney. 2017. Multi-object classification via crowdsourcing with a reject option. *IEEE Trans. Sign. Process.* 65, 4 (2017), 1068–1081.
[6] Nuno Luz, Nuno Silva, and Paulo Novais. 2015. A survey of task-oriented crowdsourcing. *Artif. Intell. Rev.* 44, 2 (2015), 187–213.
[7] Robbie T. Nakatsu, Elissa B. Grossman, and Charalambos L. Lacovou. 2014. A taxonomy of crowdsourcing based on task complexity. *J. Inf. Sci.* 40, 6 (2014), 823–834.
[8] X. Guo, H. Wang, Y. Song, and H. Gao. 2014. Brief survey of crowdsourcing for data mining. *Expert Syst. Appl.* 41, 17 (2014), 7987–7994.
[9] A. Doan, R. Ramakrishnan, and A. Y. Halevy. 2011. Crowdsourcing systems on the world-wide web. *Commun. ACM* 54, 4 (2011), 86–96.

[10] S. Badihi and A. Heydarnoori. 2017. CrowdSummarizer: Automated generation of code summaries for java programs through crowdsourcing. *IEEE Softw.* 34, 2 (2017), 71–80.

[11] G. Zou, A. Gil, and M. Tharayil. 2014. An agent-based model for crowdsourcing systems. In *Proceedings of the 2014 Winter Simulation Conference (WSC-14)*. 407–418.

[12] L. Hetmank. 2013. Components and functions of crowdsourcing systems-a systematic literature review. Retrieved from http://aisel.aisnet.org/wi2013/4.

[13] G. Goel, A. Nikzad, and A. Singla. 2014. Mechanism design for crowdsourcing markets with heterogeneous tasks. In *Proceedings of the 2nd AAAI Conference on Human Computation and Crowdsourcing (HCOMP'14)*, 77–86, Pittsburgh, Pennsylvania, USA, November 2-4, 2014.

[14] K. Osipov and G. Sukthankar. 2012. AmalgaCloud: Social network adaptation for humana and computational agent team formation. In *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering (ASE'12)*. 61–73.

[15] D. Yang, G. Xue, X. Fang, and J. Tang. 2012. Crowdsourcing to smartphones; Incentive mechanism design for mobile phone sensing. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom'12)*. 173–184.

[16] P. E. Petruzzi, J. Pitt, and D. Busquets. 2017. Electronic social capital for self-organising muti-agent systems. *ACM Trans. Auton. Adapt. Syst.* 12, 3 (2017), Article 3.

[17] B. P. Gerkey and M. J. Mataric. 2004. A formal analysis and taxonomy of task allocation in multi-robot systems. *Int. J. Robot. Res.* 23, 9 (2004), 939–954.

[18] Ross Conroy, Yifeng Zeng, Marc Cavazza, and Yingke Chen. 2015. Learning behaviors in agents systems with interactive dynamic influence diagrams. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15)*. 39–45.

[19] Y. Jiang, Y. Zhou, and W. Wang. 2013. Task allocation for undependable multiagent systems in social networks. *IEEE Trans. Parallel Distrib. Syst.* 24, 8 (2013), 1671–1681.

[20] B. T. R. Savarimuthu, M. Purvis, and M. Fleurke. 2004. Monitoring and controlling of a multi-agent based workflow system. In *Proceedings of the 2nd Workshop on Australasian Information Security, Data Mining and Web Intelligence, and Software Internationalisation (ACSW Frontiers'04)*. 127–132.

[21] Y. Jiang. 2016. A survey of task allocation and load balancing in distributed systems. *IEEE Transactions on Parallel and Distributed Systems* 27, 2, 585–599, 2016.

[22] R. Zlot and A. Stentz. 2006. Market-based multirobot coordination for complex tasks. *Int. J. Robot. Res.* 25, 1 (2006), 73–101.

[23] Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R. Jennings. 2014. Efficient crowdsourcing of unknown experts using multi-armed bandits. *Artif. Intell.* 214 (2014), 89–111.

[24] J. T. Jacques and P. O. Kristensson. 2013. Crowdsourcing a HIT: Measuring workers' pre-task interactions on microtask markets. In *Proceedings of the 1st AAAI Conference on Human Computation and Crowdsourcing (HCOMP'13)*. 86–93.

[25] Long Tran-Thanh, M. Venanzi, A. Rogers, and N. R. Jennings. 2013. Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'13)*. 901–908.

[26] A. Kittur, B. Smus, and R. Kraut. 2011. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST'11)*. 43–52.

[27] H. Jiang and S. Matsubara. 2014. Efficient task decomposition in crowdsourcing. In *Proceedings of the 17th International Conference on Principles and Practice of Multi-Agent Systems (PRIMA'14)*, LNAI Vol. 8861. 65–73.

[28] Victor R. Lesser. 1999. Cooperative multiagent systems: A personal view of the state of the art. *IEEE Trans. Knowl. Data Eng.* 11, 1 (1999), 133–142.

[29] J. Lee and L. F. Lai. 2002. A high-level petri nets-based approach to verifying task structures. *IEEE Trans. Knowl. Data Eng.* 14, 2 (2002), 316–335.

[30] Y. Jiang, J. Hu, and D. Lin. 2011. Decision making of networked multiagent systems for interaction structures. *IEEE Trans. Syst. Man Cybernet. A* 41, 6 (2011), 1107–1121.

[31] A. Garvey, M. Humphrey, and V. R. Lesser. 1993. Task interdependencies in design-to-time real-time scheduling. In *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI'93)*. 580–585.

[32] J. Yu and R. Buyya. 2006. A taxonomy of workflow management systems for grid computing. *J. Grid Comput.* 3, 3 (2006), 171–200.

[33] Peng Dai, Christopher H. Lin, and Daniel S. Weld. 2013. POMDP-based control of workflows for crowdsourcing. *Artif. Intell.* 202 (2013), 52–85.

[34] P. Tichy, P. Slechta, R. J. Staron, F. P. Maturana, and K. H. Hall. 2006. Multiagent technology for fault tolerance and flexible control. *IEEE Trans. Syst. Man Cybernet. C* 36, 5 (2006), 700–704.

[35] Gagan Goel, Afshin Nikzad, and Adish Singla. 2014. Allocating tasks to workers with matching constraints: Truthful mechanisms for crowdsourcing markets. In *Proceedings of the 23rd International Conference on World Wide Web Companion (WWW'14)*. 279–280.

[36] Sarvapali D. Ramchurn, Claudio Mezzetti, Andrea Giovannucii, Juan A. Rodriguez-Aguilar, Rajdeep K. Dash, and Nicholas R. Jennings. 2009. Trust-based mechanisms for robust and efficient task allocation in the presence of execution un-certainty. *J. Artif. Intell. Res.* 35, 1 (2009), 119–159.

[37] T. D. Huynh, N. R. Jennings, and N. R. Shadbolt. 2006. An integrated trust and reputation model for open multi-agent systems. *Auton. Agents Multi-Agent Syst.* 13, 2 (2006), 119–154.

[38] Ruggiero Cavallo, David C. Parkes, and Satinder Singh. 2012. Optimal coordinated planning amongst self-interested agents with private state. arXiv:12066820 [cs.GT].

[39] David R. Karger, Sewoong Oh, and Devavrat Shah. 2014. Budget-optimal task allocation for reliable crowdsourcing systems. *Operat. Res.* 62, 1 (2014), 1–24.

[40] L. Mo et al. 2013. Optimizing plurality for human intelligence tasks. In *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management (CIKM'13)*. 1929–1938.

[41] Han Yu, Zhiqi Shen, Chunyan Miao, and Bo An. 2013. A reputation-aware decision-making approach for improving the efficiency of crowdsourcing systems. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'13)*. 1315–1316.

[42] M. Venanzi, Alex Rogers, and Nicholas R. Jennings. 2013. Trust-based fusion of untrustworthy information in crowdsourcing applications. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'13)*. 829–836.

[43] Yu Zhang and Mihaela van derSchaar. 2012. Reputation-based incentive protocols in crowdsourcing applications. In *Proceedings of the 31st Annual IEEE International Conference on Computer Communications (INFOCOM'12)*. 2140–2148.

[44] G. Radanovic and R. Faltings. 2015. Incentive schemes for participatory sensing. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'15)*. 1081–1089.

[45] D. Zhao, X. Y. Li, and H. Ma. 2014. How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanism with budget constraint. In *Proceedings of the 33rd Annual IEEE International Conference on Computer Communications (INFOCOM'14)*. 1213–1221.

[46] Anand Kulkarni, Matthew Can, and Björn Hartmann. 2012. Collaboratively crowdsourcing workflows with turkomatic. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW'12)*. 1003–1012.

[47] H. Rahman, S. B. Roy, S. Thirumuruganathan, S. Amer-Yahia, and G. Das. 2015. Task assignment optimization in collaborative crowdsourcing. In *Proceedings of the 2015 IEEE International Conference on Data Mining (ICDM'15)*. 949–954.

[48] A. Kobren and C. H. Tan. 2015. Getting more for less: Optimized crowdsourcing with dynamic tasks and goals. In *Proceedings of the 24th International World Wide Web Conference (WWW'15)*. 592–602.

[49] C. J. Ho and J. W. Vaughan. 2012. Online task assignment in crowdsourcing markets. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI'12)*. 45–51.

[50] Bergenti Federico, Enrico Franchi, and Agostino Poggi. 2013. Agent-based interpretations of classic network models. *Comput. Math. Organiz. Theory* 19, 2 (2013), 105–127.

[51] Jiming Liu, Xiaolong Jin, and Yuanshi Wang. 2005. Agent-based load balancing on homogeneous minigrids: Macroscopic modeling and characterization. *IEEE Trans. Parallel Distrib. Syst.* 16, 7 (2005), 586–598.

[52] F. Daniel, P. Kucherbaev, C. Cappiello, B. Benatallah, and M. Allahbakhsh. 2018. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Comput. Surv.* 51, 1 (2018), Article 7.

[53] A. Tampuu, T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, and J. Aru. 2017. Multiagent cooperation and competition with deep reinforcement learning. *PLoS ONE* 12, 4 (2017), e0172395.

[54] Y. Jiang, Yifeng Zhou, and Yunpeng Li. 2015. Reliable task allocation with load balancing in multiplex networks. *ACM Trans. Auton. Adapt. Syst.* 10, 1 (2015), Article 3.

[55] Dayong Ye, Minjie Zhang, and Athanasios V. Vasilakos. 2017. A survey of self-organization mechanisms in multiagent systems. *IEEE Trans. Syst. Man Cybernet. Syst.* 47, 3 (2017), 441–461.

[56] P. Xiong, L. Zhang, and T. Zhu. 2017. Reward-based spatial crowdsourcing with differential privacy preservation. *Enterp. Inf. Syst.* 11, 10 (2017), 1500–1517.

[57] F. Zambonelli. 2011. Pervasive urban crowdsourcing: Visions and challenges. In *Proceedings of the 7th IEEE International Workshop on Pervasive Learning, Life, and Lei-sure (PerEL'11)*. 578–583.

[58] Retrieved from https://answers.yahoo.com/.

[59] Lingjie Duan, Kubo T. Sugiyama, and K. Jianwei Huang. 2012. Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing. In *Proceedings of the 2012 IEEE International Conference on Computer Communications (INFOCOM'12)*. 1701–1709.

[60] A. Biswas, D. Chander, K. Dasgupta, K. Mukherjee, M. Singh, and T. Mukherjee. 2015. PISCES: Participatory incentive strategies for effective community engagement in smart cities. In *Proceedings of the 3rd AAAI Conference on Human Computation and Crowdsourcing (HCOMP'15)*. 23–31.

[61] Y. Pan and E. Blevis. 2011. A survey of crowdsourcing as a means of collaboration and the implications of crowdsourcing for interaction design. In *Proceedings of the 2011 International Conference on Collaboration Technologies and Systems (CTS'11)*. 397–403.

[62] David R. Karger, Sewoong Oh, and Devavrat Shah. 2011. Iterative learning for reliable crowdsourcing systems. In *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS'11)*. 1953–1961.

[63] L. G. Jaimes, I. Vergara-Laurens, and M. A. Labrador. 2012. A location-based incentive mechanism for participatory sensing systems with budget constraints. In *Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications (PerCom'12)*. 103–108.

[64] Donghui Lin and Toru Ishida. 2014. Coordination of local process views in interorganizational business process. *IEICE Trans. Inf. Syst.* E97-D, 5 (2014), 1119–1126.

[65] Danny Weyns, Robrecht Haesevoets, and Alexander Helleboogh. 2010. The MACODO middleware for context-driven dynamic agent organizations. *ACM Trans. Auton. Adapt. Syst.* 5, 1 (2010), Article 3.

[66] G. Weiss. 1999. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. The MIT Press, Cambridge, MA, 1999.

[67] Reza Olfati-Saber. 2006. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. Autom. Contr.* 51, 3 (2006), 401–420.

[68] Onn Shehory and Sarit Kraus. 1998. Methods for task allocation via agent coalition formation. *Artif. Intell.* 101, 1–2 (1998), 165–200.

[69] S. D. Ramchurn, D. Huynh, and N. R. Jennings. 2004. Trust in multi-agent systems. *Knowl. Eng. Rev.* 19, 1 (2004), 1–25.

[70] Alessandro Bozzon, Marco Brambilla, Stefano Ceri, Matteo Silvestri, and Giuliano Vesci. 2013. Choosing the Right crowd: Expert finding in social networks. In *Proceedings of the 16th International Conference on Extending Database Technology (EDB'13)*. 637–648.

[71] A. Kulkarni, P. Narula, D. Rolnitzky, and N. Kontny. 2014. Wish: Amplifying creative ability with expert crowds. In *Proceedings of the 2nd AAAI Conference on Human Computation and Crowdsourcing (HCOMP'14)*. 112–120.

[72] Yukino Baba, Hisashi Kashima, Kei Kinoshita, Goushi Yamaguchi, and Yosuke Akiyoshi. 2014. Leveraging non-expert crowdsourcing workers for improper task detection in crowdsourcing marketplaces. *Expert Syst. Appl.* 41, 6 (2014), 2678–2687.

[73] Theodoros Lappas, Kun Liu, and Evimaria Terzi. 2009. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*. 127–132.

[74] Anirban Majumder, Samik Datta, and K. V. M. Naidu. 2012. Capacitated team formation problem on social networks. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*. 1005–1013.

[75] S. Jain, B. Narayanaswamy, and Y. Narahari. 2014. A multiarmed. bandit incentive mechanism for crowdsourcing demand response in smart grids. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'14)*. 721–727.

[76] W. Hu, L. Liu, and G. Feng. 2017. Output consensus of heterogeneous linear multi-agent systems by distributed event-triggered/self-triggered strategy. *IEEE Trans. Cybernet.* 47, 8 (2017), 1914–1924.

[77] S. Assadi, J. Hsu, and S. Jabbari. 2015. Online assignment of heterogeneous tasks in crowdsourcing markets. In *Proceedings of the 3rd AAAI Conference on Human Computation and Crowdsourcing (HCOMP'15)*. 2–11.

[78] M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich. 2015. Soylent: A word processor with a crowd in-side. *Commun. ACM* 58, 8 (2015), 313–322.

[79] J. Ferber and O. Gutknecht. 1998. A meta-model for the analysis and design of organizations in multi-agent systems. In *Proceedings of the 3rd International Conference on Multi Agent Systems (ICMAS'98)*. 128–135.

[80] Aniket Kittur, Ed H. Chi, and Bongwon Suh. 2008. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI 2008 Conference on Human Factors in Computing Systems (CHI'08)*. 453–456.

[81] J. Chamberlain. 2014. Groupsourcing: Distributed problem solving using social networks. In *Proceedings of the 2nd AAAI Conference on Human Computation and Crowdsourcing (HCOMP'14)*. 22–29.

[82] Y. Jiang and J. C. Jiang. 2014. Understanding social networks from a multiagent perspective. *IEEE Trans. Parallel Distrib. Syst.* 25, 10 (2014), 2743–2759.

[83] Y. Jiang and Z. Huang. 2012. The rich get richer: Preferential attachment in the task allocation of cooperative networked multiagent systems with resource caching. *IEEE Trans. Syst. Man Cybernet. A* 42, 5 (2012), 1040–1052.

[84] Qing Liu, Tie Luo, Ruiming Tang, and Stéphane Bressan. 2015. An efficient and truthful pricing mechanism for team formation in crowdsourcing markets. In *Proceedings of the 2015 IEEE International Conference on Communications (ICC'15)*. 567–572.

[85] Mehdi Kargar, Aijun An, and Morteza Zihayat. 2012. Efficient bi-objective team formation in social networks. In *Proceedings of the 2012 European Conference on Ma-chine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD'12)*, LNCS Vol. 7524. 483–498.

[86] Markus Rokicki, Sergej Zerr, and Stefan Siersdorfer. 2015. Groupsourcing: Team competition designs for crowdsourcing. In *Proceedings of the 24th International Conference on World Wide Web (WWW'15)*. 906–915.

[87] A. Singla et al. 2014. Building collaborative teams: A local navigation-based approach. Retrieved from http://research.microsoft.com/en-us/um/people/ryenw/papers/SinglaNIPS2014.pdf.

[88] Ece Kamar and Eric Horvitz. 2012. Incentives for truthful reporting in crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'12)*. 1329–1330.

[89] W. Feng, Z. Yan, H. Zhang, K. Zeng, Y. Xiao, and Y. T. Hou. 2018. A survey on security, privacy and trust in mobile crowdsourcing. *IEEE IoT Journal*, 10.1109/JIOT.2017.2765699, 2018.

[90] Vamshi Ambati, Stephan Vogel, and Jaime Carbonell. 2012. Collaborative workflow for crowdsourcing translation. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW'12)*. 1191–1194.

[91] Jiang Yang, Lada A. Adamic, and Mark S. Ackerman. 2008. Crowdsourcing and knowledge sharing: Strategic user behavior on taskcn. In *Proceedings of the 9th ACM Conference on Electronic Commerce (EC'08)*. 246–255.

[92] Aniket Kittur, Susheel Khamkar, Paul André, and Robert E. Kraut. 2012. CrowdWeaver: Visually managing complex crowd work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW'12)*. 1033–1036.

[93] Peter Stone and Manuela Veloso. 1999. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artif. Intell.* 110, 2 (1999), 241–273.

[94] Guido Sautter and Klemens Böhm. 2013. High-throughput crowdsourcing mechanisms for complex tasks. *Soc. Netw. Anal. Min.* 3, 4 (2013), 873–888.

[95] Y. Jiang and J. Jiang. 2009. Contextual resource negotiation-based task allocation and load balancing in complex software systems. *IEEE Trans. Parallel Distrib. Syst.* 20, 5 (2009), 641–653.

[96] Shibo He, Dong-Hoon Shin, Junshan Zhang, and Jiming Chen. 2014. Toward optimal allocation of location dependent tasks in crowdsensing. In *Proceedings of the 33rd Annual IEEE International Conference on Computer Communications (INFOCOM'14)*. 745–753.

[97] Cen Chen, Shih-Fen Cheng, Archan Misra, and Hoong Chuin Lau. 2015. Multi-agent task assignment for mobile crowdsourcing under trajectory uncertainties. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS'15)*. 1715–1716.

[98] Blazej Bulka, Matthew Gaston, and Marie desJardins. 2007. Local strategy learning in networked multi-agent team formation. *Auton. Agents Multi-Agent Syst.* 15, 1 (2007), 29–45.

[99] Alessandro Bozzon, Ilio Catallo, Eleonora Ciceri, Piero Fraternali, Davide Martinenghi, and Marco Tagliasacchi. 2012. A framework for crowdsourced multimedia processing and querying. In *Proceedings of the 21st International Conference on World Wide Web on Workshop on CrowdSearch (WWW'12)*. 42–47.

[100] Panos Ipeirotis, Greg Little, and Thomas W. Malone. 2014. Composing and analyzing crowdsourcing workflows. Retrieved from http://www.ipeirotis.com/wp-content/uploads/2014/03/main.pdf.

[101] Christopher H. Lin, Daniel S. Weld, and others. 2012. Dynamically switching between synergistic workflows for crowdsourcing. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI'12)*. 87–93.

[102] Cheong Ha Park, KyoungHee Son, Joon Hyub Lee, and Seok-Hyung. 2013. Crowd vs. Crowd: Large-scale cooperative design through open team competition. In *Proceedings of the 16th ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW'13)*.

[103] Luca de Alfaro, V. Polychronopoulos, and M. Shavlovsky. 2015. Reliable aggregation of boolean crowdsourced tasks. In *Proceedings of the 3rd AAAI Conference on Hu-man Computation and Crowdsourcing (HCOMP'15)*. 42–51.

[104] D. Yue, G. Yu, D. Shen, and X. Yu. 2014. A weighted aggregation rule in crowdsourcing systems for high result accuracy. In *Proceeding of the 12th International Conference on Dependable, Autonomic and Secure Computing (DASC'14)*. 265–270.

[105] Mihai Georgescu and Xiaofei Zhu. 2014. Aggregation of crowdsourced labels based on worker history. In *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS'14)*.

[106] T. Matsui, Y. Baba, T. Kamishima, and H. Kashima. 2013. Crowdsourcing quality control for item ordering tasks. *Human Computation and Crowdsourcing: Works in Progress and Demonstration Abstracts, AAAI Technical Report CR-13-01*, pp. 52–53.

[107] J. Hui, A. Glenn, R. Jue, E. Gerber, and S. Dow. 2015. Using anonymity and communal efforts to improve quality of crowdsourced feedback. In *Proceedings of the 3rd AAAI Conference on Human Computation and Crowdsourcing (HCOMP'15)*. 72–81.

[108] Ming Yin and Yiling Chen. 2015. Bonus or not? learn to reward in crowdsourcing. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15)*. 201–207.

[109] D. Dipalantino and M. Vojnovic. 2009. Crowdsourcing and all-pay auctions. In *Proceedings of the 10th ACM Conference on Electronic Commerce (EC'09)*. 119–128.

[110] Benjamin Satzger, Harald Psaier, Daniel Schall, and Schahram Dustdar. 2013. Auction-based crowdsourcing supporting skill management. *Inf. Syst.* 38, 4 (2013), 547–560.

[111] Maria Daltayanni, Luca de Alfaro, and Panagiotis Papadimitriou. 2015. WorkerRank: Using employer implicit judgments to infer worker reputation. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining (WSDM'15)*. 263–272.

[112] M. Kokkodis and P. G. Ipeirotis. 2013. Have you done anything like that? Predicting performance using inter-category reputation. In *Proceedings of the 6th ACM Inter-national Conference on Web Search and Data Mining (WSDM'13)*. 435–444.

[113] D. R. Karger, S. Oh, and D. Shah. 2011. Budget-optimal crowdsourcing using low-rank matrix approximations. In *Proceedings of the 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton'11)*. 284–291.

[114] R. K. Ganti, F. Ye, and H. Lei. 2011. Mobile crowdsensing: Current stage and future challenges. *IEEE Commun. Mag.* 49, 11 (2011), 32–39.

[115] A. Anagnostopoulos, L. Becchetti, A. Fazzone I. Mele, and M. Riondato. 2015. The importance of being expert: Efficient max-fining in crowdsourcing. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (SIGMOD'15)*. 983–998.

[116] J. G. T. Zañudo, G. Yang, and R. Albert. 2017. Structure-based control of complex networks with nonlinear dynamics. *Proc. Natl. Acad. Sci. U.S.A.* 114, 28 (2017), 7234–7239.

[117] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz. 2006. Market-based multirobot coordination: A survey and analysis. *Proc. IEEE* 94, 7 (2006), 1257–1270.

[118] C. Sahin et al. 2017. A multi-agent based approach to dynamic scheduling with flexible processing capabilities. *J. Intell. Manufactur.* 28, 8 (2017), 1827–1845.

[119] S. B. Roy, I. Lykourentzou, S. Thirumuruganathan, S. Amer-Yahia, and G. Das. 2015. Crowds, not drones: Modeling human factors in interactive crowdsourcing. Retrieved September 21, 2015 from https://hal.inria.fr/hal-00923542/.

[120] G. Meng and D. V. Dimarogonas. 2017. Task and motion coordination for heterogeneous multiagent systems with loosely coupled local tasks. *IEEE Trans. Autom. Sci. Eng.* 14, 2 (2017), 797–808.

[121] D. Grosu and A. T. Chronopoulos. 2004. Algorithmic mechanism design for load balancing in distributed systems. *IEEE Trans. Syst. Man Cybernet. B* 34, 1 (2004), 77–84.

[122] A. Kittur, J. V. Nickerson, E. Gerber, A. Shaw, J. Zimmerman, M. Lease, and J. Horton. 2013. The future of crowd work. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW'13)*. 1301–1318.

[123] A. J. Quinn and B. B. Bederson. 2011. Human computation: A survey and taxonomy of a growing field. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'11)*. 1403–1402.

[124] K. Mao, L. Capra, M. Harman, and Y. Jia. 2017. A survey of the use of crowdsourcing in software engineering. *J. Syst. Softw.* 126 (2017), 57–84.

[125] J. Jiang, B. An, Y. Jiang, and D. Lin. 2017. Context-aware reliable crowdsourcing in social networks. (unpublished).

Received July 2017; revised February 2018; accepted April 2018