Tsinghua Science and Technology

Volume 23 | Issue 6 Article 7

2018

Truthful Mechanism for Crowdsourcing Task Assignment

Yonglong Zhang
School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China. College of Information Engineering, Yangzhou University, Yangzhou 225127, China.

Haiyan Qin
College of Information Engineering, Yangzhou University, Yangzhou 225127, China.

Bin Li
College of Information Engineering, Yangzhou University, Yangzhou 225127, China.

Jin Wang
College of Information Engineering, Yangzhou University, Yangzhou 225127, China.

Sungyoung Lee
Department of Computer Engineering, Kyung Hee University, Suwon 449-701, Korea.

See next page for additional authors

Recommended Citation

Yonglong Zhang, Haiyan Qin, Bin Li et al. Truthful Mechanism for Crowdsourcing Task Assignment. Tsinghua Science and Technology 2018, 23(6): 645-659.

TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 02/10 pp 645–659
DOI: 10.26599 / TST.2018.9010064
Volume 23, Number 6, December 2018

Truthful Mechanism for Crowdsourcing Task Assignment

Yonglong Zhang, Haiyan Qin, Bin Li, Jin Wang, Sungyoung Lee, and Zhiqiu Huang*

Abstract: As an emerging "human problem solving strategy", crowdsourcing has attracted much attention where requesters want to employ reliable workers to complete specific tasks. Task assignment is an important branch of crowdsourcing. Most existing studies in crowdsourcing have not considered self-interested individuals' strategy. To guarantee truthfulness, auction has been regarded as a promising method to charge the requesters for the tasks completed and reward the workers for performing the tasks. In this study, an online task assignment scenario is considered where each worker has a set of experienced skills, whereas a specific task is budget-constrained and requires one or more skills. In this scenario, the crowdsourcing task assignment was modeled as a reverse auction where the requesters are buyers and the workers are sellers. Three incentive mechanisms, namely, Truthful Mechanism for Crawdsourcing-Vickrey-Clarke-Grove (TMC-VCG), TMC-Simple Task (ST) for a simple task case, and TMC-Complex Task (CT) for a complex task case are proposed. Here, a simple task case means that the requester asks for a single skill, and a complex task case means that the requester asks for multiple skills. The related properties of each of the three mechanisms are determined theoretically. Moreover, the truthfulness is verified, and other performances are evaluated by extensive simulations.

Key words: crowdsourcing; task assignment; auction; truthfulness

1 Introduction

Crowdsourcing has emerged as an efficient "human problem solving" strategy for where the requesters ask for certain workers to complete the tasks. Howe[1] introduced the term "crowdsourcing". It is defined as the method of outsourcing a specific task to an undefined set of people, instead of assigning the task to designated employees. In general, crowdsourcing can be regarded as a method of matching requesters, who post tasks with a limited budget, with skilled workers, who perform the tasks in a timely manner to earn money[2]. The crowdsourcing platform acts as a mediator between the requesters and workers. There are many crowdsourcing platforms, such as Amazon Mechanical Turk (AMT)[3], a typical crowdsourcing platform where requesters post human intelligence tasks to online workers. AMT offers only "microtasks" with small rewards, such as social science experiments and process photos, whereas TopCoder[4], a crowdsourcing marketplace for software development, offers larger tasks such as designing a software.

In this study, the task assignment in crowdsourcing was modeled as a reverse auction, where the requesters are buyers, workers are sellers, and crowdsourcing platform serves as an auctioneer. Here, the terms requester and

• Yonglong Zhang and Zhiqiu Huang are with School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China. E-mail: ylzhang@yzu.edu.cn; zqhuang@nuaa.edu.cn.
• Yonglong Zhang, Haiyan Qin, Bin Li, and Jin Wang are with College of Information Engineering, Yangzhou University, Yangzhou 225127, China. E-mail: ylzhang@yzu.edu.cn; 13092006508@163.com; lb@yzu.edu.cn; wangjin@nuist.edu.cn.
• Sungyoung Lee is with Department of Computer Engineering, Kyung Hee University, Suwon 449-701, Korea. E-mail: sylee@khu.ac.kr.
* To whom correspondence should be addressed.
Manuscript received: 2017-02-23; revised: 2017-04-17; accepted: 2017-04-20

buyer, worker and seller, and crowdsourcing platform and auctioneer are interchangeable. The human problem solving process in crowdsourcing is shown in Fig. 1. In each round of auction, the crowdsourcing platform first screens the online players as the requesters and workers arrive and leave over time. The requesters submit tasks with limited budgets, and the workers offer their skills with comparable costs to the auctioneer. After collecting all the information from the buyers and sellers, the auctioneer conducts the Winner Determination and Pricing (WDP) process. Then, the winning workers are matched with the requesters, perform tasks, and receive payments from the platform charged from the winning requesters. Finally, the requesters and workers rate each other according to their performance.

Crowdsourcing has several issues such as designing tasks, finding crowd, and quality control. In this study, we focused on the following challenges: First, task assignment can be classified into offline and online cases based on the accessibility of participants[5]. In offline cases, the requesters/workers submit their ask/bid simultaneously at the beginning, and there are no arriving entities. Then, the crowdsourcing platform provides the assignment results with all the available information. Conversely, in online version, the ask/bid profiles are sent sequentially when requesters/workers arrive. Online cases are complex because the platform must make decisions on the fly without a prior knowledge of players arriving in the future. In the task allocation stage, not only efficient task assignment, but also the quality of task assignment should be considered. The task assignment process can be improved by gathering and exploiting knowledge on quality[6]. The quality of workers indicates task satisfaction when the tasks are completed by them. Considering malicious requesters, the quality of requesters should also be considered. Naturally, rational requesters prefer to hand over the tasks to workers with higher qualities. Thus, the quality of players is the second challenge in online task assignment.

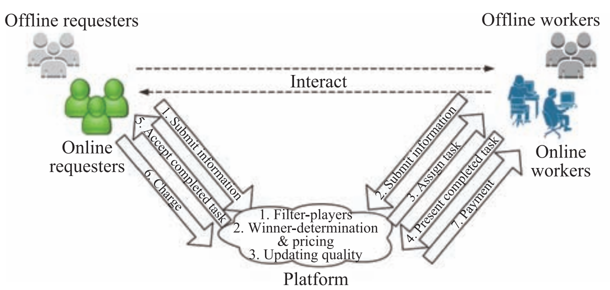


Fig. 1 Model of task assignment in crowdsourcing.

The third challenge is the main focus of this paper, this originates from self-serving players who may attempt to increase utility by misreporting their bid or ask. Hassan and Curry[6] categorized the dimensions of dynamic assignment to optimization, constraints, learn, and context. Slivkins and Vanghan[2] classified specific directions into five aspects, namely, adaptive task assignment, dynamic procurement, repeated principal-agent problem, reputation systems, and one common theme: the exploration-exploitation tradeoff. Although incentive models were analyzed in Ref. [5], the incentive mechanism still has not been considered in the literature. All the above surveys did not consider the players' strategy. In other words, the payment in crowdsourcing is rashly decided by the workers' ask or the requesters' bid. To guarantee truthfulness, auction has been introduced into many fields such as cooperative communication[7], cloud resource provisioning[8, 9], and even crowdsourcing market[10]. Undoubtedly, auction[11] is an efficient way to hinder manipulation.

In this paper, we propose three incentive mechanisms to solve the challenges mentioned above. The main contributions of this study are as follows.

- In this study, task assignment in crowdsourcing is cast as a reverse auction free from market manipulation. An online scenario was considered where no prior information is available about the players arriving in the future. Moreover, the qualities of requesters and workers were considered simultaneously. The task was divided into two cases: a simple task and a complex task. For a simple task, a Vickrey-Clarke-Grove (VCG)-like scheme, Truthful Mechanism for Crowdsourcing (TMC)-VCG was designed. To overcome the imbalance of the budget and high time complexity in TMC-VCG, TMC-Simple Task (ST) that guarantees all the desired properties is proposed. For a complex task, TMC-Complex Task (CT) requiring the collective efforts of multiple workers was designed. TMC-CT uses a threshold price to decide the payment to guarantee truthfulness.
- TMC-ST and TMC-CT have good economic properties including truthfulness, individual rationality, and budget balance, whereas TMC-VCG satisfies two of them except the budget balance because of VCG-like payment. The simulation results verify the truthfulness of the proposed mechanisms and then evaluate other desirable performances.

The remainder of this paper is structured as follows: Section 2 provides a brief review of related studies. Section 3 describes the model system in detail, definitions, and design objectives. Sections 4–6 present three incentive mechanisms, respectively, and then analyze the related properties, specifically the truthfulness. Section 7 presents the simulation results. Section 8 provides the conclusions of this study.

2 Related Studies

As a promising paradigm, auction is applied in various fields. In this study, we focused on incentive mechanisms in crowdsourcing market. The related studies can be categorized into two groups: (1) task assignment without incentive mechanism and (2) crowdsourcing using auction scheme. The related studies are shown in Table 1, and the major differences in mechanisms between the previous studies and this study are also shown.

2.1 Crowdsourcing without incentive mechanisms

In a Spatial Crowdsourcing (SC) scenario, Hassan and Curry[6] proposed a distance-reliability ratio algorithm for dynamic task assignment that aims to maximize task reliability and minimize travel costs. Considering the quality and cost, Distance-Reliability Ratio (DRR) algorithm takes away the truthfulness of players. To solve multiskill SC (MS-SC) problems, efficient approximation approaches, namely, greedy, g-divide-and-conquer, and cost-model-based adaptive algorithms are proposed in Ref. [12]. Without regard to reward division, their MS-SC problem assigns tasks to workers, so that the required skills of tasks can be covered. The problem of Ref. [13] is an example of online knapsack problem that aims to maximize the number of assigned tasks with a fixed overall budget. Although they ignore the mechanism design, they claim that all their online algorithms satisfy truthfulness. In a large-scale crowdsourcing scenario such as crowdsourcing software engineering, Yue et al.[14] proposed a search-based approach to solve optimization problems. Failing to consider the price mechanism, the search-based approach simply finds a fit match between the task submitter and virtual team members.

All the above studies did not consider incentive mechanisms; therefore, cheating still exists in their cases. Many players may benefit from the design, and the profit of the platform may be damaged.

Table 1 Comparison between previous studies and this study.

Reference	Online	Offline	Quality	Truthful
[6]	√	×	√	×
[12]	√	×	×	×
[13]	√	√	×	×
[14]	×	×	×	×
[15]	×	×	×	√
[16]	×	×	×	√
[17]	√	×	×	√
[18]	√	×	√	√
[19]	√	√	×	√
This paper	√	×	√	√

2.2 Crowdsourcing with incentive mechanism

Liu et al.[15] proposed four incentive mechanisms to build a valid team to complete a complex task. Among the mechanisms, optimal and VCG-like mechanisms are not computationally efficient, and the greedy mechanism is untruthful. Therefore, TruTeam[15] is proposed that satisfies all the desired properties. TruTeam uses Myerson's theorem[20] to ensure the truthfulness. Another incentive mechanism using Myerson's theorem is reported in Ref. [16]. Zhang et al.[16] studied three models of crowdsourcing, namely, Single-Requester Single-Bid (SS-Model), Single-Requester Multiple-Bid (SM-Model), and Multiple-Requester Multiple-Bid (MM-Model), and designed an incentive mechanism for each of these models. Considering the time and location information of tasks, an incentive mechanism for mobile crowdsourcing is proposed in Ref. [17], where a VCG-like payment scheme is applied to guarantee the truthfulness. In Ref. [18], a Multiarmed Bandit (MAB) problem is added to incentive mechanism in an online crowdsourcing scenario. The budget is divided into exploration and exploitation budgets for the sake of learning. Motivated from the VCG mechanism design and greedy approach of the M-sensing algorithm, SMART is presented in Ref. [19] for an offline case. Besides, SMART algorithm can be easily used for an online case.

3 Problem Formulation

3.1 System model

In this study, the task assignment problem is cast as a reverse auction that runs in a time-slotted fashion denoted by $[T] = \{1, 2, ..., T\}$, where T is a particularly large integer. Each time slot $t \in [T]$ is a predefined time segment such as an hour or a day. At the beginning of slot t, several requesters and workers arrive in the platform. In this model, it was assumed that there are τ types of skills, and the skill profile is denoted by $S = \{s_1, s_2, ..., s_\tau\}$. Next, the three entities in our model are characterized.

Requester: $R_t = \{r_1, r_2, ..., r_n\}$ denotes a set of requesters in slot t. Each requester $r_i \in R_t$ submits her bid denoted by $D_i = \{S_i^r, b_i, q_{i,t}, It_i, Ft_i\}$ to the platform. $S_i^r = \{s_i^1, s_i^2, ..., s_i^\tau\}$ denotes the skill requirement of r_i where $s_i^k \in \{0,1\}$ indicates if the task posted by r_i needs skill s_k or not. b_i is r_i's claimed maximal price which she is willing to offer for the given task. $\tilde{b_i}$ is used to denote the true valuation of r_i for employing the demanded workers. $q_{i,t}$ represents the quality of r_i in slot t. r_i reaches in $It_i \in [T]$ and will leave in $Ft_i \in [T]$. Ft_i is the latest time r_i can tolerate to allocate workers. The bids of all requesters are denoted as $D_t = \{D_1, D_2, ..., D_n\}$.

Worker: Let $A_t = \{a_1, a_2, ..., a_m\}$ denote the set of workers in slot t. Each worker $a_j \in A_t$ submits his ask indicated by $C_j = \{\boldsymbol{S}_j^a, It_j, Ft_j\}$ to the platform. $\boldsymbol{S}_j^a = \{< s_j^k, c_j^k, q_{j,t}^k >_{k=1,2,...,\tau}\}$ is a skill vector where $s_j^k \in \{0,1\}$ represents whether a_j possesses k-th skill or not. c_j^k is a submitted minimum acceptable reward for providing skill s_j^k. Because of the selfish worker, the true cost for supplying s_j^k is denoted by $\tilde{c_j^k}$, which is not necessarily equal to c_j^k. $q_{j,t}^k$ is the quality of a_j for supplying skill s_j^k, which is updated as t progresses. If $s_j^k = 0$, $c_j^k = 0$. Similarly, a_j will submit the arrival time It_j and the leaving time Ft_j to the platform. Let $C_t = \{C_1, C_2, ..., C_m\}$ represent the ask profile of all workers in A_t.

Platform: After gathering D_t and C_t, the platform will decide the outcome of the auction, which is denoted by $\Phi(t) = \{W_t^r, W_t^a, \boldsymbol{P}^r, P^a, \sigma\}$. $W_t^r \in R_t$ and $W_t^a \in A_t$ are the winning requesters and winning workers in slot t, respectively. $\boldsymbol{P}^r = \{P_1^r, P_2^r, ..., P_n^r\}$ is the price vector collected from requesters, where P_i^r is the price charged from r_i. Similarly, the payment reward to seller a_j is P_j^a, and $P^a = \{P_1^a, P_2^a, ..., P_m^a\}$ is the payment to all the workers. Most importantly, the auctioneer decides the mapping function $\sigma : \{i : r_i \in R_t\} \to \{j : a_j \in A_t\}$, i.e., if $j = \sigma(i)$, r_i will employ a_j to perform the task.

In this model, workers only submit the ask of a single skill, and similarly the quality is evaluated singly. To cope with gathered skills, the overall cost and quality are imported. $\boldsymbol{H} = \{h_1, h_2, ..., h_\tau\}$ represents the weight vector of S, where h_k is the workload of skill s_k. If all the skills share the same workload, $h_1 = h_2 = \cdots = h_\tau$. If $\exists_{k=1,2,...,\tau} s_j^k \geqslant s_i^k$, the overall cost oC_j^i and overall quality oQ_j^i of a_j for r_i can be calculated as follows:

$$oC_j^i = \frac{\sum_{k=1}^{\tau} h_k s_i^k c_j^k s_j^k}{\sum_{k=1}^{\tau} h_k s_i^k s_j^k} \tag{1}$$

$$oQ_j^i = \frac{\sum_{k=1}^{\tau} h_k s_i^k q_j^k s_j^k}{\sum_{k=1}^{\tau} h_k s_i^k s_j^k} \tag{2}$$

Therefore, the utility of winning requester $r_i \in R_t$ is the difference between the true valuation and payment:

$$U_i^r = \begin{cases} \tilde{b_i} - P_i^r, & \text{if } r_i \in W_t^r \text{ ;} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Similarly, the utility of winning worker $a_j \in A_t$ is the reward from the auctioneer minus the cost of supplying the skills $r_{\sigma^{-1}(j)}$ requests:

$$U_j^a = \begin{cases} P_j^a - o\widetilde{C_j^{\sigma^{-1}(j)}}, & \text{if } a_j \in W_t^a \text{ ;} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Here, $o\widetilde{C_j^{\sigma^{-1}(j)}}$ is the true overall cost corresponding to $oC_j^{\sigma^{-1}(j)}$.

3.2 Definition of concepts

This section introduces several economic properties desired to achieve and some concepts used in this study.

Definition 1 (Truthfulness). An auction is truthful if and only if his/her true valuation is his/her dominant strategy. Namely, the players cannot improve their utility by misreporting their bid or ask. In our auction, this indicates that for $\forall r_i \in R_t$, if $b_i = \tilde{b_i}$, U_i^r is maximized; for $\forall a_j \in A_t$, if $oC_j^{\sigma^{-1}(j)} = o\widetilde{C_j^{\sigma^{-1}(j)}}$, U_j^a is maximized.

Truthfulness is the most crucial property in auction theory. Because exposing a true bid or ask procures the highest utility, no rational buyer or seller will cheat any more. As a result, the trade can be free from market manipulation. In addition, irrespective of other's behavior, players will simply apply their truth-telling strategy, so that the system will reach an equilibrium.

Definition 2 (Individual rationality). In an individual rational auction. Therefore, the players will submit the truth so the winner gains a non-negative utility, and the loser's utility is zero. In other words, no winning requester is charged more than her bid, and no winning worker is paid less than his ask. This signifies $P_i^r \leqslant b_i$ and $P_j^a \geqslant oC_j^{\sigma^{-1}(j)}$ for $r_i \in W_t^r$ and $a_j \in W_t^a$, respectively.

Definition 3 (Budget balance). Budget balance means that the auctioneer gains a non-negative utility, i.e., the money collected from the requesters is not less than the reward defrayed to worker. That is to say, $\sum_{r_i \in W_t^r} P_i^r \geqslant \sum_{a_j \in W_t^a} P_j^a$.

Definition 4 (Computational efficiency). The auction mechanism is computationally efficient if and only if it could be executed within a polynomial time.

Definition 5 (Social welfare). Social welfare is defined as the aggregate utility of all buyers, sellers, and auctioneer, i.e., the difference between the sum of winning requesters' bid and the sum of winning workers' ask.

$$SW = \sum_{r_i \in W_t^r} \tilde{b}_i - \sum_{a_j \in W_t^a} oC_j^{\tilde{\sigma}^{-1}(j)} \tag{5}$$

An auction is system efficient if the utility of all players is optimized.

3.3 Design objective

On one hand, as a traditional auction, our mechanism targets three economic properties, namely, truthfulness, individual rationality, and budget balance. One the other hand, the objective of our mechanism is to achieve system efficiency. The optimization problem is formalized as follows:

$$\text{Objective: maximize} \sum_{i=1}^{n} \sum_{j=1}^{m} \left(\tilde{b}_i - o\tilde{C}_j^i \right) \cdot x_{i,j},$$

subject to

$$x_{i,j} \in \{0,1\},\ \forall r_i \in R_t,\ \forall a_j \in A_t,\ t \in [T] \tag{6}$$

$$\sum_{j=1}^{m} x_{i,j} \geqslant 0,\ \forall r_i \in R_t;\ \sum_{i=1}^{n} x_{i,j} \in \{0,1\}, \forall a_j \in A_t \tag{7}$$

The objective is restricted by Formulas (6) and (7). $x_{i,j} = \{0,1\}$ in Formula (6) indicates whether r_i will recruit a_j or not. Formula (7) is the matching constraint for all the proposed schemes. The first formula means that each winning requester can be matched with more than one worker, while the loser is assigned to no worker. More concretely, in a simple task case, the winning requester is assigned only one worker, while more than one worker in a complex task case. The second formula shows that each worker can be assigned to at most one requester.

4 TMC-VCG Mechanism

In this section, we propose a VCG-like truthful mechanism for crowdsourcing, TMC-VCG for short. VCG scheme is the most well-known optimal allocation scheme that can guarantee truthfulness. Applied in a simple task case, TMC-VCG offers sufficient consideration to quality to improve the assignment process. Our theoretical analysis proves that TMC-VCG achieves truthfulness, individual rationality, and computational efficiency.

4.1 Design details

TMC-VCG consists of three steps: filter-players, WDP, and update-quality. In the filter-players phase, the online players are updated. Winning requesters, winning workers, the charge from the requesters, the payment to the workers, and the matching between the requester and worker are decided in the WDP phase. After the task is accomplished by the worker, the quality of players will be updated in the update-quality phase. Our VCG-like mechanism is given in Algorithm 1.

Filter-players: At the beginning of each time slot, the online players are screened, and then the remaining procedure is carried out. Simply, if the termination time of players is later than the current time, he/she is online, and vice versa. The requesters in current time slot t consist of R_{t-1}, unallocated but still online requesters after the assignment in $t-1$, and nR_t, the newcomer requesters in slot t. Similarly, A_t is the aggregation of A_{t-1} and nA_t.

WDP: In the context of a simple task, the requester demands a single skill to complete the task, i.e., for each $r_i \in R_t$, $\sum_{k=1}^{\tau} s_i^k = 1$. As a variant of the bid, virtual bid is applied as reported in Refs. [21, 22]. Combining the bid and ask with quality, virtual bid and virtual ask are employed in TMC-VCG and defined as follows.

Definition 6 (Virtual bid). The virtual bid of r_i is the product of bid and quality, i.e., $vB_i = b_i \cdot q_i$.

Definition 7 (Virtual ask). The virtual ask of a_j is the product of ask and quality, i.e., $vC_j = c_j^k \cdot q_j^k$. Here, it was assumed that s_j^k is exactly $r_{\sigma^{-1}(j)}$ demand.

Algorithm 1 TMC-VCG

Input: $R_{t-1}, A_{t-1}, nR_t, nA_t$.
Output: $\Phi(t)$.

{Phase 1: filter-players}
$R_t \leftarrow R_{t-1} \cup nR_t$
$A_t \leftarrow A_{t-1} \cup nA_t$
{Phase 2: WDP}
Create a weighted complete bipartite graph
$G \leftarrow (R_t, A_t, WE)$,where $we_{i,j} \in WE$ is calculated as follows:
$we_{i,j} = \begin{cases} vB_i - vC_j, & \text{if } \forall_{k=1,2,\ldots,\tau} s_j^k \geqslant s_i^k; \\ 0, & \text{otherwise.} \end{cases}$
$(W_t^r, W_t^a, \sigma) \leftarrow MWM(G)$
for $r_i \in W_t^r$ **do**
  $G_{-r_i} \leftarrow \left(R_{-r_{i,t}}, A_t, WE_{-r_i}\right)$
  $\left(W_t^{r'}, W_t^{a'}, \sigma'\right) \leftarrow MWM(G_{-r_i})$
  $P_i^r = b_i - \left(vSW^\star - vSW_{-r_i}^\star\right)/q_i$
  $R_t \leftarrow R_t \setminus \{r_i\}$
end for
for $a_j \in W_t^a$ **do**
  $G_{-a_j} \leftarrow \left(R_t, A_{-a_{j,t}}, WE_{-a_j}\right)$
  $\left(W_t^{r'}, W_t^{a'}, \sigma'\right) \leftarrow MWM(G_{-r_i})$
  $P_j^a = c_j^k + \left(vSW^\star - vSW_{-a_j}^\star\right)/q_j^k$
  $A_t \leftarrow A_t \setminus \{a_j\}$
end for
{Phase 3: update-quality}
for $r_i \in W_t^r$ **do**
  $q_{i,t+1} \leftarrow Q(q_{i,t})$
end for
for $a_j \in W_t^a$ **do**
  if a_j contributes $s_j^k\ (k = 1,2,\ldots,\tau)$ **then**
    $q_{j,t+1}^k \leftarrow Q\left(q_{i,t}^k\right)$
  end if
end for
Return $\{W_t^r, W_t^a, P^r, P^a, \sigma\}$

Definition 8 (Virtual social welfare). The virtual social welfare is the sum of the difference between the winning requester's virtual bid and the corresponding winning worker's virtual ask, formalized as vSW:

$$vSW = \sum_{r_i \in W_t^r} (vB_i - vC_{\sigma(i)}) \tag{8}$$

A VCG scheme is normally based on assignment with the maximum social welfare, while the assignment in TMC-VCG scheme achieves the maximum virtual social welfare. In the WDP stage, first a weighted bipartite graph with two sets of nodes representing the requesters and workers is created. The weight of each edge, $we_{i,j} \in WE$ is calculated as follows:

$$we_{i,j} = \begin{cases} vB_i - vC_j, & \text{if } \forall_{k=1,2,\ldots,\tau} s_j^k \geqslant s_i^k; \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

Then, the Maximum Weighted Matching (MWM) algorithm[23] aiming at maximizing vSW is selected, i.e., $vSW^\star$. From line 7 to line 18, MWM algorithm is restarted for calculating $vSW_{-r_i}^\star$ when r_i is removed from auction and $vSW_{-a_j}^\star$ when a_j is removed from auction. The charge of winning requesters and the payment of winning workers are shown as follows:

$$P_i^r = b_i - \left(vSW^\star - vSW_{-r_i}^\star\right)/q_i \tag{10}$$

$$P_j^a = c_j^k + \left(vSW^\star - vSW_{-a_j}^\star\right)/q_j^k \tag{11}$$

Here, it is assumed that s_j^k is exactly the skill that requester $r_{\sigma^{-1}(j)}$ needs.

Update-quality: Once the task is over, the winning requesters and workers evaluate each other. The dynamic assignment while learning is known as the adaptive assignment problem in Ref. [2]. Despite the learning process, our work emphasizes the solution to player's strategy rather than online learning. Hence, we will not discuss the learning process, but adopt the existing adaptivity algorithms. MAB is always studied to solve the adaptive assignment problem. The function $Q(\cdot)$ in lines 21 and 25 can employ the algorithms reported in Refs. [18, 24, 25].

4.2 Analysis of desirable properties

In the following, it is proved that TMC-VCG has the properties of truthfulness, individual rationality, and computational efficiency.

Theorem 1 TMC-VCG achieves truthfulness.

Proof It has been shown in Ref. [26] that a VCG-like auction is truthful. Because of limited space, only the truthfulness of requester is proved. The utility of r_i is computed as follows:

$$\begin{aligned} U_i^r = & \tilde{b_i} - P_i^r = \\ & \tilde{b_i} - b_i + \left(vSW^\star - vSW_{-r_i}^\star\right)/q_i = \\ & \tilde{b_i} - b_i + \Big[b_i \cdot q_i - c_{\sigma(i)}^k \cdot q_{\sigma(i)}^k + \\ & \sum_{r_l \in W_t^r \backslash r_i} \left(b_l \cdot q_l - c_{\sigma(l)}^k \cdot q_{\sigma(l)}^k\right) - vSW_{-r_i}^\star\Big]/q_i = \\ & \tilde{b_i} + \Big[-c_{\sigma(i)}^k \cdot q_{\sigma(i)}^k + \\ & \sum_{r_l \in W_t^r \backslash r_i} \left(b_l \cdot q_l - c_{\sigma(l)}^k \cdot q_{\sigma(l)}^k\right) - vSW_{-r_i}^\star\Big]/q_i = \\ & \Big[\tilde{b_i} \cdot q_i - c_{\sigma(i)}^k \cdot q_{\sigma(i)}^k + \\ & \sum_{r_l \in W_t^r \backslash r_i} \left(b_l \cdot q_l - c_{\sigma(l)}^k \cdot q_{\sigma(l)}^k\right) - vSW_{-r_i}^\star\Big]/q_i = \\ & \left(v\tilde{SW}^\star - vSW_{-r_i}^\star\right)/q_i. \end{aligned}$$

Based on the above reduction, $U_i^r = (v\tilde{SW}^\star - vSW_{-r_i}^\star)/q_i$ where $v\tilde{SW}^\star$ means virtual social welfare when r_i bids $b_i = \tilde{b_i}$. Because $vSW_{-r_i}^*$ is independent of r_i, U_i^r is maximized when the matching is $vSW^\star$, i.e., when r_i bids $b_i = \tilde{b_i}$. Therefore, no requester can improve its utility by bidding untruthfully; thus, the truthfulness of requesters is proved. ■

Theorem 2 TMC-VCG achieves individual rationality.

Proof The charge of the winning requester r_i is $P_i^r = b_i - \left(vSW^\star - vSW_{-r_i}^\star\right)/q_i$. Among the parameters, $vSW^\star - vSW_{-r_i}^\star$ is obviously non-negative, so that $\left(vSW^\star - vSW_{-r_i}^\star\right)/q_i$ is also non-negative. Therefore, P_i^r is less than b_i. The individual rationality for workers can be proved in the same way, we omit here. ■

Theorem 3 TMC-VCG is computationally efficient.

Proof We primarily discuss the time complexity in the WDP stage. It takes $O(mn)$ to create a weighted complete bipartite graph. The time complexity of the MWM algorithm is $O(\max\{n,m\}^3)$[27]. The pricing stages for requesters and workers are $O(n \cdot \max\{n,m\}^3)$ and $O(m \cdot \max\{n,m\}^3)$, respectively. Therefore, TMC-VCG is computationally efficient. ■

5 TMC-ST Mechanism

In TMC-VCG, the imbalance of budget is unacceptable to the auctioneer. Moreover, to overcome the time complexity of exponential growth in TMC-VCG, we propose an iterative incentive mechanism for a simple task assignment, called TMC-ST. Our theoretical analysis shows that TMC-ST achieves truthfulness, individual rationality, budget balance, and computational efficiency.

5.1 Details of auction

Similarly, TMC-ST consists of three steps: filter-players, WDP, and update-quality; they are described in Algorithm 2. Among them, filter-players and update-quality are the same as those in TMC-VCG; therefore, we omit the illustration of the two stages here. Now, we present WDP in details.

WDP: In each slot $t \in [T]$, we first sort online requesters by $\text{Rank}(r_i)$ in a nonincreasing order. As described in line 6, $\text{Rank}(r_i)$ is related to urgency and quality. Simply, for $r_i \in R_t$, the smaller $Ft_i - t$ is, the more urgent the task is. The larger the $q_{i,t}$ is, the more trustworthy the requester is, the larger the qualityRank(r_i) is. Here, $\alpha \in [0,1]$ is a tune parameter between urgency and quality. The requesters in the top of sorted sequence have a higher distribution priority. If some requesters have the same rank, they could be selected randomly or reordered by requester id. In this study, the default way is to reorder those requesters by requester id. We go through the reordered requesters successively and search the matched workers sequentially. We regard the allocation to each requester as a single-round auction. For each $r_i \in R_t$, A^i is the set of unallocated workers whose skills satisfy the skill request of r_i. Here are two cases according to the number of requesters in A^i.

- If there is only one worker in A^i, whether r_i can be a winner depends on mb_i which is the median bid of the latest requesters who are online or expiring throughout the whole auction process and share the same skill vector with r_i. If the constraint in line 12 is satisfied, the trade between requester and the worker is a success; otherwise, it is a failure.
- If $|A^i| > 1$, there are more than one online workers satisfying r_i's skill requirement. As described in line 19, the workers in A^i are ranked in a nonincreasing order of reliability, which is defined as his quality q_j^k by ask c_j^k, i.e., $\text{rel}(a_j) = q_j^k / c_j^k$. It selects the worker a_{j1} with the highest reliability. $p_{i,j1}$, the payment to the candidate a_{j1} is calculated in line 20. If $b_i \geqslant p_{i,j1}$, the deal between r_i and a_{j1} is effective, or else r_i and a_{j1} will not be the winner.

Algorithm 2 TMC-ST

Input: $R_{t-1}, A_{t-1}, nR_t, nA_t$.
Output: $\Phi(t)$.
1: {Phase 1: filter-players}
2: $R_t \leftarrow R_{t-1} \cup nR_t$
3: $A_t \leftarrow A_{t-1} \cup nA_t$
4: {Phase 2: WDP}
5: **for** $r_i \in R_t$ **do**
6: $\quad \text{Rank}(r_i) = \alpha \cdot \text{urgencyRank}(r_i) + (1-\alpha) \cdot \text{qualityRank}(r_i)$
7: **end for**
8: Sort all the requesters in R_t to get an ordered list, $R_t = \{r_{i1}, r_{i2}, \ldots, r_{in}\}$ such that $\text{Rank}(r_{i1}) \geqslant \text{Rank}(r_{i2}) \geqslant \ldots \geqslant \text{Rank}(r_{in})$
9: **for** $i \leftarrow i1$ to in **do**
10: $\quad A^i = \{a_j \mid \forall_{k=1,2,\ldots,\tau} s_j^k \geqslant s_i^k, a_j \in A_t\}$
11: $\quad$ **if** $|A^i| = 1$ **then**
12: $\quad\quad$ **if** $b_i \geqslant mb_i$ and $c_j^k \leqslant mb_i$ **then**
13: $\quad\quad\quad j = \sigma(i), W_t^r \leftarrow W_t^r \cup \{r_i\}, W_t^a \leftarrow W_t^a \cup \{a_j\}$
14: $\quad\quad\quad P_i^r = P_j^a = mb_i$
15: $\quad\quad\quad R_t \leftarrow R_t \setminus \{r_i\}, A_t \leftarrow A_t \setminus \{a_j\}$
16: $\quad\quad$ **end if**
17: $\quad$ **else if** $|A^i| > 1$ **then**
18: $\quad\quad$ Compute $\text{rel}(a_j) = \frac{q_j^k}{c_j^k}$ for each $a_j \in A^i$
19: $\quad\quad$ Sort all the workers in A^i to get an ordered list, $A^i = \{a_{j1}, a_{j2}, \ldots\}$ such that $\text{rel}(a_{j1}) \geqslant \text{rel}(a_{j2}) \geqslant \ldots$
20: $\quad\quad p_{i,j1} = \frac{q_{j1}^k}{q_{j2}^k} \cdot c_{j2}^k$
21: $\quad\quad$ **if** $b_i \geqslant p_{i,j1}$ **then**
22: $\quad\quad\quad j1 = \sigma(i), W_t^r \leftarrow W_t^r \cup \{r_i\}, W_t^a \leftarrow W_t^a \cup \{a_{j1}\}$
23: $\quad\quad\quad P_i^r = P_{j1}^a = p_{i,j1}$
24: $\quad\quad\quad R_t \leftarrow R_t \setminus \{r_i\}, A_t \leftarrow A_t \setminus \{a_{j1}\}$
25: $\quad\quad$ **end if**
26: $\quad$ **end if**
27: **end for**
28: {Phase 3: update-quality}
29: **for** $r_i \in W_t^r$ **do**
30: $\quad q_{i,t+1} \leftarrow Q(q_{i,t})$
31: **end for**
32: **for** $a_j \in W_t^a$ **do**
33: $\quad$ **if** a_j contributes s_j^k $(k = 1, 2, \ldots, \tau)$ **then**
34: $\quad\quad q_{j,t+1}^k \leftarrow Q\left(q_{i,t}^k\right)$
35: $\quad$ **end if**
36: **end for**
37: Return $\{W_t^r, W_t^a, P^r, P^a, \sigma\}$

5.2 Analysis of desirable properties

In this section, we prove that TMC-ST achieves three economic properties, namely, truthfulness, individual rationality, and budget balance. In addition, TMC-ST is computationally efficient.

Theorem 4 TMC-ST achieves truthfulness.

Before proving the truthfulness of TMC-ST, we first prove that winner-determination is monotonic, and pricing is bid-independent. To differentiate truthful and untruthful cases, we added tilde in the notations for the truthful case.

(1) Monotonic winner determination

Lemma 1 If r_i wins by bidding $\tilde{b_i}$, she can also win by bidding $b_i > \tilde{b_i}$, given other parameters fixed.

Proof Algorithm 2 shows that the members in A^i are related to skill request of r_i. Hence, A^i is the same when bidding b_i or $\tilde{b_i}$. There are two subcases according to the number of elements in A^i.

- $|A^i| = 1$: When there is only one worker in A^i, r_i wins by bidding $\tilde{b_i}$, indicating that $\tilde{b_i} \geqslant mb_i$. Accordingly, r_i can also win by bidding $b_i > \tilde{b_i} \geqslant mb_i$.
- $|A^i| > 1$: Similarly, r_i wins by bidding $\tilde{b_i} \geqslant p_{i,j1}$; therefore, she can also win by bidding $b_i > \tilde{b_i} \geqslant p_{i,j1}$.

Thus, Lemma 1 has been proved. ■

Lemma 2 If a_j wins by asking $\tilde{c_j^k}$, he can also win by bidding $c_j^k < \tilde{c_j^k}$, given other parameters fixed.

Proof We assume that the skill s_j^k is the same that $r_{\sigma^{-1}(j)}$ demands. In brief, when a_j decreases the ask to c_j^k, the constraint $c_j^k \leqslant mb_i$ and $c_{j1}^k \leqslant p_{i,j1}$ are still satisfied whether $|A^i| = 1$ or $|A^i| > 1$. Therefore, the worker will still win when decreasing his ask. ■

(2) Bid-independent pricing

Lemma 3 Given other parameters fixed, if r_i wins by bidding $\tilde{b_i}$ or b_i, she is charged by the same price, i.e., $\tilde{P_i^r} = P_i^r$.

Proof As mentioned in Lemma 1, A^i is unchanged by bidding $\tilde{b_i}$ or b_i. Moreover, mb_i and $p_{i,j1}$ are irrelevant to r_i's bid; therefore, the price of requester is bid-independent. ■

Lemma 4 Given other parameters fixed, if a_j wins by bidding $\tilde{c_j^k}$ or c_j^k, he is paid the same price, i.e., $\tilde{P_j^a} = P_j^a$.

Proof Similarly, mb_i or $p_{i,j1}$ is irrelevant to a_j's ask, so $\tilde{P_j^a} = P_j^a = mb_i$ or $\tilde{P_j^a} = P_j^a = p_{i,j1}$ is independent of a_j's ask. ■

Lemma 5 TMC-ST is truthful for requesters.

Proof As mentioned above, the distribution to a requester at a time is considered as a single auction in which the truthfulness is evaluated. To compare the requester's utility when biding truthfully and untruthfully, we discuss the truthfulness of requesters from four cases. For $r_i \in R_t$,

Case 1: In slot t, if r_i loses no matter bids $\tilde{b_i}$ or b_i, then $\tilde{U_i^r} = U_i^r = 0$.

Case 2: If r_i wins by bidding $\tilde{b_i}$ but loses by bidding b_i, clearly $\tilde{U_i^r} \geqslant U_i^r = 0$.

Case 3: If r_i wins when bidding b_i but loses by bidding $\tilde{b_i}$, the above tells that A^i is invariant no matter r_i bids b_i or $\tilde{b_i}$. There are two subcases according to the number of elements in A^i:

- $|A^i| = 1$: In this subcase, the reason r_i loses by bidding $\tilde{b_i}$ is that $\tilde{b_i} < mb_i$. Thus, r_i will win by raising his bid till $b_i \geqslant mb_i$ where mb_i is bid-independent. The above shows that $P_i^r = mb_i \geqslant \tilde{b_i}$.
- $|A^i| > 1$: In the same way, a_{j1} has not been assigned to r_i because $\tilde{b_i} < p_{i,j1}$. Therefore, r_i must change her bid to $b_i \geqslant P_i^r = p_{i,j1} > \tilde{b_i}$.

Hence, $U_i^r = \tilde{b_i} - P_i^r < \tilde{U_i^r} = 0$.

Case 4: If r_i wins whether bidding b_i or $\tilde{b_i}$, we know that P_i^r is bid-independent from Lemma 3. Hence, $U_i^r = \tilde{U_i^r} = \tilde{b_i} - P_i^r$.

Overall, $\tilde{U_i^r} \geqslant U_i^r$, indicating that TMC-ST is truthful for requesters. ■

Lemma 6 TMC-ST is truthful for workers.

Proof In the same way, we discuss the truthfulness of workers from four cases. We assume that s_j^k is the same $r_{\sigma^{-1}(j)}$ needs. For $a_j \in A_t$,

Case 1: If a_j fails to be allocated to $r_{\sigma^{-1}(j)}$ by asking c_j^k or $\tilde{c_j^k}$, $U_j^a = \tilde{U_j^a} = 0$.

Case 2: If a_j is allocated to $r_{\sigma^{-1}(j)}$ successfully by bidding $\tilde{c_j^k}$, but loses by bidding c_j^k, then $\tilde{U_j^a} \geqslant U_j^a = 0$.

Case 3: If a_j wins by asking c_j^k, but loses when asking $\tilde{c_j^k}$, we discuss two conditions:

- $|A^{\sigma^{-1}(j)}| = 1$: The reason a_j loses in this condition is $\tilde{c_j^k} > mb_{\sigma^{-1}(j)}$; therefore, a_j must reduce his ask until $c_j^k < mb_{\sigma^{-1}(j)}$. Therefore, $U_j^a = P_j^a - \tilde{c_j^k} = mb_{\sigma^{-1}(j)} - \tilde{c_j^k} < 0 = \tilde{U_j^a}$.
- $|A^{\sigma^{-1}(j)}| > 1$: There are two potential reasons why a_j loses in this condition. One reason is that $j = j_1$, but $\tilde{c_j^k} > p_{i,j_1}$. Hence, a_j must reduce the ask to c_j^k, i.e., $c_j^k < p_{i,j_1} = P_j^a < \tilde{c_j^k}$. Another reason is that $j \neq j_1$, i.e., $\mathrm{rel}(a_j)$ is not the largest in $A^{\sigma^{-1}(j)}$. Therefore, a_j must reduce his ask c_j^k till $\mathrm{rel}(a_j)$ is the first member in $A^{\sigma^{-1}(j)}$, i.e., $\mathrm{rel}(a_j) \geqslant \mathrm{rel}(a_{j_1})$. When a_j asks for $\tilde{c_j^k}$, $\mathrm{rel}(a_{j_1}) \geqslant \ldots \geqslant \mathrm{rel}(a_j)$, i.e., $\dfrac{q_{j_1}^k}{c_{j_1}^k} \geqslant \dfrac{q_j^k}{\tilde{c_j^k}}$; therefore, $\tilde{c_j^k} \geqslant \dfrac{q_j^k}{q_{j_1}^k} \cdot c_{j_1}^k$. Hence, when a_j changes his ask to c_j^k, $P_j^a = \dfrac{q_j^k}{q_{j_1}^k} \cdot c_{j_1}^k \leqslant \tilde{c_j^k}$.

From the two conditions, $U_j^a = P_j^a - \tilde{c_j^k} \leqslant 0 = \tilde{U_j^a}$.

Case 4: a_j wins by asking c_j^k or $\tilde{c_j^k}$. Lemma 4 shows that the payment will not change no matter a_j asks for c_j^k or $\tilde{c_j^k}$. Thus, $U_j^a = \tilde{U_j^a} = P_j^a - \tilde{c_j^k}$.

Theorem 4 can be proved together by Lemma 5 and Lemma 6, i.e., TMC-ST is truthful for requesters and workers. ■

Theorem 5 TMC-ST achieves individual rationality.

Proof In Algorithm 2, no matter $|A^{\sigma^{-1}(j)}| = 1$ or $|A^{\sigma^{-1}(j)}| > 1$, line 12 or 21 is built as the constraint of individual rationality. For $r_i \in W_t^r$, $P_i^r = mb_i \leqslant b_i$ or $P_i^r = p_{i,j_1} \leqslant b_i$, exhibiting individual rationality for requesters. As well, for $a_j \in W_t^a$, $P_j^a = mb_i \geqslant c_j^k$ or $P_j^a = p_{i,j_1} \geqslant c_j^k$ also indicates individual rationality for workers. ■

Theorem 6 TMC-ST achieves budget balance.

Proof Considering this one-to-one matching between W_t^r and W_t^a, $|W_t^r| = |W_t^a|$. The clearing price satisfies $P_i^r = P_j^a$ no matter $|A^{\sigma^{-1}(j)}| = 1$ or $|A^{\sigma^{-1}(j)}| > 1$. Therefore, it can be easily shown that

$$\sum_{r_i \in W_t^r} P_i^r - \sum_{a_j \in W_t^a} P_j^a = \sum_{r_i \in W_t^r} \left(P_i^r - P_{\sigma(i)}^a\right).$$

This completes the proof. ■

Theorem 7 TMC-ST is computationally efficient.

Proof In the WDP stage, we first sort the requesters with a complexity of $O(n\log n)$. Each round of for-loop for requesters takes $O(m\log m)$ time. In total, Algorithm 2 has a time complexity of $O(n\log n + mn\log m)$ that is computationally efficient. ■

6 TMC-CT Mechanism

In this section, we consider the scenario where a complex task may require the collective efforts of multiple workers. In general, many complex tasks cannot be completed with a single skill such as software development. We propose TMC-CT for complex tasks requiring one or more workers' skill combination. Our theoretical proof shows that TMC-CT guarantees truthfulness, individual rationality, and budget balance, moreover a reachable time complexity.

6.1 Design details

Similar to TMC-VCG and TMC-ST, TMC-CT consists of three steps: filter-players, WDP, and update-quality. The difference lies in the second step. We employ a critical value to address the manipulation problem. The detailed WDP process is as follows.

WDP: As illustrated in Algorithm 3, we allocate the requesters in a greedy manner. First, like TMC-ST, the online requesters are sorted as $\mathrm{Rank}(r_i)$ declines. In line 10, A^i is the set of workers who satisfy r_i's one or more skills. If the skills of r_i have not been covered and the workers in A^i are not empty, we iteratively select the worker a_j with the minimum cost per marginal contribution per quality as the candidate. Here, we provide the definition of marginal contribution.

Definition 9 (Marginal contribution) a_j's marginal contribution to r_i, Δ^i_j, is relevant to the r_i's uncovered skills that a_j can cover if selected into S_i, i.e.,

$$\Delta^i_j = \sum_{k=1}^{\tau} s^k_j \cdot s^k_i \cdot h_k \tag{12}$$

It should be mentioned that in each iteration, oC^i_j, oQ^i_j, and Δ^i_j will be reassigned after r_i's uncovered skills are updated in line 24.

Next, it decides the payment to the selected worker. If a_j is added to the team successfully, he will be paid the threshold price[15] that a_j will not win if he asks higher than that. Let us see how this mechanism rebuilds a team without a_j's participation. It selects a_l from the set $A^i\backslash\{S_i \bigcup a_j\}$ with the minimum cost per marginal contribution per quality. The above process is repeated until the budget is over, or the task can be completed

Algorithm 3 TMC-CT

Input: $R_{t-1}, A_{t-1}, nR_t, nA_t$.
Output: $\Phi(t)$.
1: {Phase 1: filter-players}
2: $R_t \leftarrow R_{t-1} \cup nR_t$
3: $A_t \leftarrow A_{t-1} \cup nA_t$
4: {Phase 2: WDP}
5: **for** $r_i \in R_t$ **do**
6: $\quad \mathrm{Rank}(r_i) = \alpha \cdot \mathrm{urgencyRank}(r_i) + (1-\alpha) \cdot \mathrm{qualityRank}(r_i)$
7: **end for**
8: Sort all the requesters in R_t to get an ordered list, $R_t = \{r_{i1}, r_{i2}, \ldots, r_{in}\}$ such that $\mathrm{Rank}(r_{i1}) \geqslant \mathrm{Rank}(r_{i2}) \geqslant \ldots \geqslant \mathrm{Rank}(r_{in})$
9: **for** $i \leftarrow i1$ to in **do**
10: $\quad A^i = \{a_j \mid \exists s^k_j \geqslant s^k_i, k = 1, 2, \ldots, \tau, a_j \in A_t\}$
11: $\quad K_i \leftarrow \varnothing$
12: $\quad$ **while** $A^i \neq \varnothing$ and the skills of r_i have not been covered **do**
13: $\quad\quad a_j \leftarrow \mathrm{argmin}_{a_j \in A^i \backslash K_i} \dfrac{oC^i_j}{\Delta^i_j \cdot oQ^i_j}$
14: $\quad\quad A' \leftarrow A^i \backslash \{K_i \bigcup \{a_j\}\}$
15: $\quad\quad T \leftarrow K_i$
16: $\quad\quad$ **while** $b_i \geqslant P^a_j$ and $\Delta^i_j \neq 0$ **do**
17: $\quad\quad\quad a_l \leftarrow \mathrm{argmin}_{a_l \in A' \backslash T} \dfrac{oC^i_l}{\Delta^i_l \cdot oQ^i_l}$
18: $\quad\quad\quad P^a_j = \max\left\{\dfrac{oC^i_l}{\Delta^i_l \cdot oQ^i_l} \cdot \Delta^i_j \cdot oQ^i_j, P^a_j\right\}$
19: $\quad\quad\quad T \leftarrow T \bigcup \{l\}$
20: $\quad\quad$ **end while**
21: $\quad\quad$ **if** $b_i \geqslant P^a_j$ **then**
22: $\quad\quad\quad K_i \leftarrow K_i \bigcup \{a_j\}, b_i \leftarrow b_i - P^a_j$
23: $\quad\quad\quad \sigma(j) = i$
24: $\quad\quad\quad$ update the uncovered skills in S^r_i
25: $\quad\quad$ **else**
26: $\quad\quad\quad A^i \leftarrow A^i \backslash \{a_j\}$
27: $\quad\quad$ **end if**
28: $\quad$ **end while**
29: $\quad$ **if** the skills of r_i have been covered **then**
30: $\quad\quad P^r_i = \sum_{a_j \in K_i} P^a_j$
31: $\quad\quad W^r_t \leftarrow W^r_t \bigcup \{r_i\}, W^a_t \leftarrow W^a_t \bigcup K_i$
32: $\quad\quad R_t \leftarrow R_t \backslash \{r_i\}, A_t \leftarrow A_t \backslash K_i$
33: $\quad$ **end if**
34: **end for**
35: {Phase 3: update-quality}
36: **for** $r_i \in W^r_t$ **do**
37: $\quad q_{i,t+1} \leftarrow Q(q_{i,t})$
38: **end for**
39: **for** $a_j \in W^a_t$ **do**
40: $\quad$ **if** a_j contributes $s^k_j\ (k = 1, 2, \ldots, \tau)$ **then**
41: $\quad\quad q^k_{j,t+1} \leftarrow Q\left(q^k_{i,t}\right)$
42: $\quad$ **end if**
43: **end for**
44: Return $\{W^r_t, W^a_t, P^r, P^a, \sigma\}$

without a_j's participation. As shown in line 18, the largest value throughout the entire iterative process is selected as the threshold price for a_j. In addition, a_j wins if and only if the requester's remaining value is no less than P^a_j. The charge of the winning requester is the sum of payment to the members in her group.

6.2 Analysis

In this section, we prove that TMC-CT satisfies truthfulness, individual rationality, and budget balance.

Theorem 8 TMC-CT achieves truthfulness.

Before proving the truthfulness of TMC-CT, we first introduce Myerson's well-known theorem.

Theorem 9 For players, the auction mechanism is truthful if[20]

• Monotone allocation: If a_j is allocated successfully by asking c_j^k $\left(s_j^k \neq 0\right)$, he will also be a winner by asking $c_j^{k'} < c_j^k$.

• Critical value: Worker a_j will not win the auction if he asks more than the critical value; in other words, the threshold price is the highest price he could ask.

Proof It is obvious that the allocation rule is monotone because if a_j wins, he will also be selected by asking a lower price, leading to a smaller cost per marginal contribution per quality.

When computing the payment to worker a_j, we select a new team without the participation of a_j. a_l is selected from $A^i \backslash \{S_i \bigcup \{a_j\}\}$ with the minimum $\dfrac{oC_l^i}{\Delta_l^i \cdot oQ_l^i}$. Therefore,

$$\frac{oC_j^i}{\Delta_j^i \cdot oQ_j^i} \leqslant \frac{oC_l^i}{\Delta_l^i \cdot oQ_l^i},$$

$$oC_j^i \leqslant \frac{oC_l^i}{\Delta_l^i \cdot oQ_l^i} \cdot \Delta_j^i \cdot oQ_j^i.$$

Otherwise, if $\dfrac{oC_j^i}{\Delta_j^i \cdot oQ_j^i} > \dfrac{oC_l^i}{\Delta_l^i \cdot oQ_l^i}$, a_l will be selected before a_j according to our allocation rule. Therefore, the payment to worker a_j is temporarily equal to this value.

$$P_j^a = \frac{oC_l^i}{\Delta_l^i \cdot oQ_l^i} \cdot \Delta_j^i \cdot oQ_j^i.$$

Although a_j can be selected as a winner, P_j^a may not be the highest price a_j can report. Because $T \bigcup \{a_l\}$ may not cover all the skill requirement of r_i. Thus, we continue to process the following iteration until the skills of r_i have been covered, or r_i cannot afford the price. Finally, the payment is equal to the threshold price described in line 18, i.e.,

$$P_j^a = \max\nolimits_{a_l \in A' \backslash T} \left\{ \frac{oC_l^i}{\Delta_l^i \cdot oQ_l^i} \cdot \Delta_j^i \cdot oQ_j^i, P_j^a \right\}.$$

Notably, oC_l^i, Δ_l^i, and oQ_l^i are updated by including a new team member. Because of the threshold price, if a_j asks more than P_j^a, a_j will be replaced after the last selected worker; in that way, he will not be selected to perform the task. The above discussion proves the critical value.

Monotone allocation and critical value are satisfied; thus, the truthfulness for workers is proved.

The reason r_i fails to be allocated is that the skills of r_i cannot be covered by the remaining workers or limited budget. Therefore, if the requester with a limited budget wants to be the winner, she has to raise her bid till greater than the team's ask. As a result, the requester will gain a negative utility. Therefore, the truthfulness for requesters is achieved. ■

Theorem 10 TMC-CT achieves individual rationality.

Proof In line 21, we check if $b_i \geqslant P_j^a$, i.e., the payment for worker a_j is less than r_i's budget. b_i will be updated to $b_i - P_j^a$ as including a new worker to the team, but the constraints $b_i \geqslant P_j^a$ will always be satisfied. After all the iteration, b_i's initial value is still larger than $P_i^r = \sum_{a_j \in K_i} P_j^a$, i.e., $U_i^r = b_i - P_i^r \geqslant 0$. Thus, the proof of individual rationality for requesters is completed.

From the payment rule for workers, $oC_j^i \leqslant P_j^a = \max_{a_l \in A' \backslash T} \left\{ \dfrac{oC_l^i}{\Delta_l^i \cdot oQ_l^i} \cdot \Delta_j^i \cdot oQ_j^i, P_j^a \right\}$. Thus, $U_j^a = P_j^a - oC_j^i \geqslant 0$. Therefore, the individual rationality of workers is achieved. ■

Theorem 11 TMC-CT achieves budget balance.

Proof The above payment rule shows that $P_i^r = \sum_{a_j \in K_i} P_j^a$, i.e., the charge of r_i is the sum of the payment to the members in the team. Therefore, $\sum_{r_i \in W_t^r} P_i^r = \sum_{a_j \in W_t^a} P_j^a$, i.e., the budget balance of TMC-CT reaches. ■

Theorem 12 TMC-CT is computationally efficient.

Proof The time complexity of sorting requester needs $O(n \log n)$ time. It takes $O(m)$ to select the worker with the minimal cost per contribution per quality. Deciding the payment for the selected worker takes $O(m\tau)$. Because there are n requesters, the time complexity of Algorithm 3 is $O(n \log n + nm^2\tau)$. This is computationally efficient. ■

7 Performance Evaluation

7.1 Simulation settings

In this section, we simulate a crowdsourcing platform where the requesters and workers arrive and leave on the fly. The three incentive mechanisms, namely, TMC-VCG, TMC-ST, and TMC-CT are implemented in this platform. We first verify the truthfulness for the three mechanisms. Then, we evaluate the performance of mechanisms for a simple task and complex task. Finally, we compare the number of transactions and running time of mechanisms. For requesters, the number of transactions means that the number of budget-limited tasks whose skill requirements are satisfied by assigned workers. For workers, the number of transactions means the number of workers successfully allocated to a requester.

The number of time slots is varied from 15 to 60, and the default value is 30. We assume that there are 12 to 15 requesters and workers that arrive at the platform and leave in any later time slot. The number of skills can be varied from 2 to 10. For TMC-CT, the size of skill vector exceeds 2 to ensure team formation. The quality of players is randomly distributed over (0, 1]. The bid and ask for different mechanisms are set in different intervals. For TMC-VCG and TMC-ST, the bid and ask are randomly selected in the interval (0, 10]. However, in view of team formation, the bid of requesters in TMC-CT is randomly distributed over (10, 15], while the ask of workers for a single skill is in the interval (0, 5]. The tune parameter α of requester rank is fixed at 0.7.

All the simulations were run on a Windows PC with a 2.94 GHz Intel$^{\mathrm{R}}$ Core$^{\mathrm{TM}}$2 Duo CPU and 2 GB memory. Each indicated data in the figures is the average result of 100 independent instances in each setting.

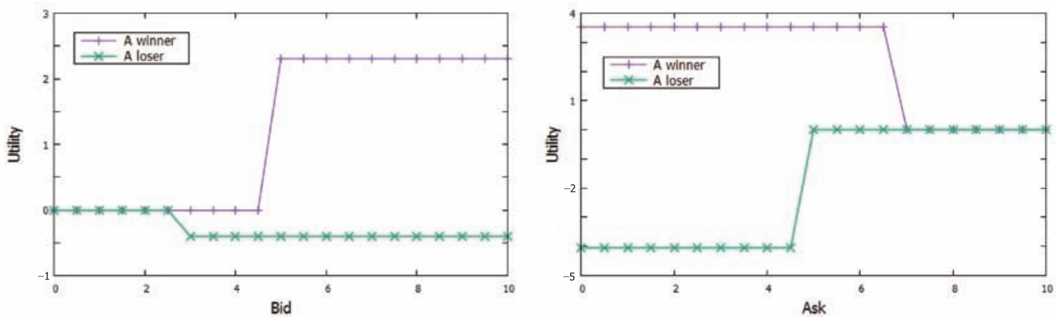
7.2 Truthfulness

To verify the truthfulness of mechanisms, we randomly select one winner and one loser and then examine how their utilities change when they bid or ask a different value. Figure 2a shows that a randomly selected winning requester bids truthfully with $b_i = \tilde{b_i} = 7.07$ and achieves utility 2.29. The utility with a truthful bid is the highest among all the possible bids. The winning requester will fail and gain no utility when bidding untruthfully with lower bids. Similarly, the loser case in Fig. 2a indicates when a losing requester raises her bid to win the auction, she will gain a negative utility. As shown in Fig. 2b, the winning worker achieves a positive utility if he asks truthfully, but loses if he attempts to increase his ask. In the same manner, the losing worker achieves zero utility when asking truthfully, but wins with a negative utility. The truthfulness for TMC-ST and TMC-CT can be evaluated in the same manner as shown in Figs. 3 and 4, respectively.

7.3 Performance evaluation for simple task assignment

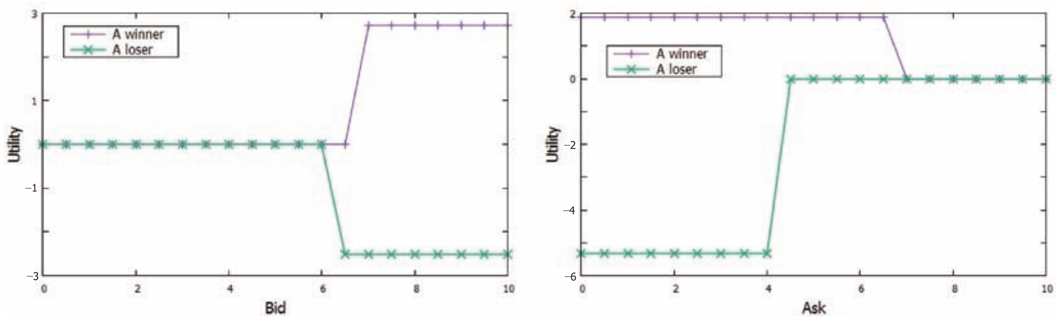
Here, we introduce the optimal allocation for a single task in the offline situation, i.e., OPTimal (OPT) ST. OPT-ST borrowed from the optimal offline mechanism reported in Ref. [17] can be described as follows: First, based on the information of requesters and workers, a weighted bipartite graph is constructed. Second, the MWM is employed to carry out the allocation between requesters and workers. Finally, VCG-like scheme is used to decide the price and guarantee truthfulness.

Figure 5a evaluates the performance of social welfare



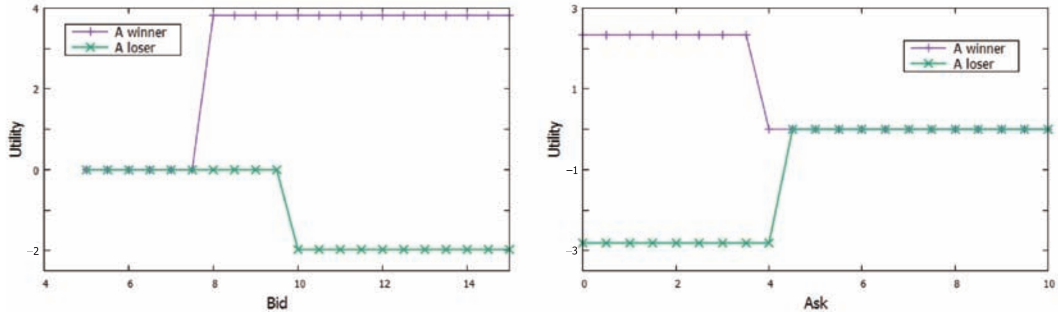
(a) Requester $r_i \in R_t$ (b) Worker $a_j \in A_t$

Fig. 2 Truthfulness of requesters and workers in TMC-VCG.



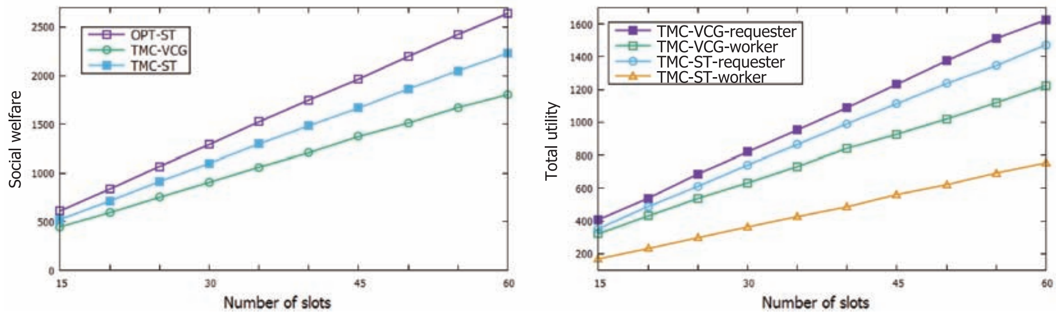
(a) Requester $r_i \in R_t$ (b) Worker $a_j \in A_t$

Fig. 3 Truthfulness of requesters and workers in TMC-ST.



(a) Requester $r_i \in R_t$ (b) Worker $a_j \in A_t$

Fig. 4 Truthfulness of requesters and workers in TMC-CT.



(a) Social welfare (b) Total utility of requester and worker

Fig. 5 Performance for simple task assignment.

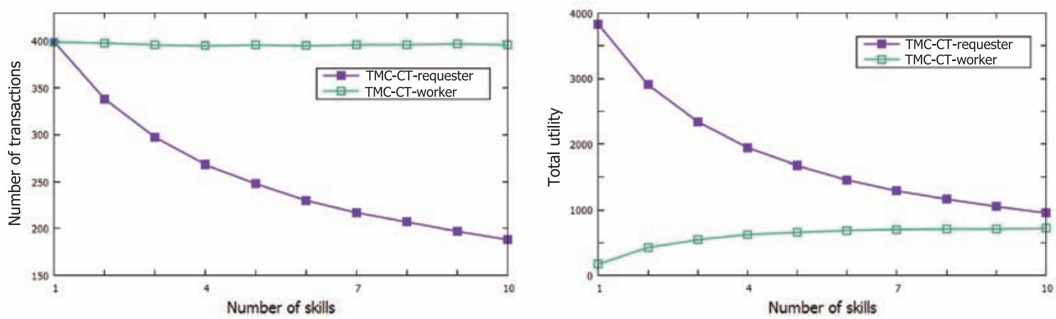
as the number of slots increases. The OPT-ST scheme whose objective is to maximize social welfare achieves the highest social welfare compared to TMC-ST and TMC-VCG schemes. Normally, a VCG-like scheme will gain a higher social welfare, while in this study, TMC-ST achieves a higher social welfare than TMC-VCG. This is because the social welfare of TMC-VCG is harmed by using a virtual bid and virtual ask to the assignment and price. Figure 5b shows that the utility of all players linearly increases as the slots expand. TMC-VCG is always superior to TMC-ST in the total utility no matter requester or worker. This is because TMC-VCG is a global optimal method, whereas TMC-ST is a local optimal solution.

7.4 Performance evaluation for complex task assignment

In this section, we evaluate the performance of TMC-CT. Figure 6 shows that the total utility of a requester decreases as the number of skills increases. This is because the demand of requesters will be more difficult to be satisfied with larger skill combinations. The total utility of a worker increases with a stable transaction count. This is because with the increase in the number of skills, the closer cooperation between workers will improve the utility of individuals.

7.5 Comparison of three mechanisms

In this section, we compare the number of transactions of



(a) Number of transactions in TMC-CT (b) Total utility in TMC-CT

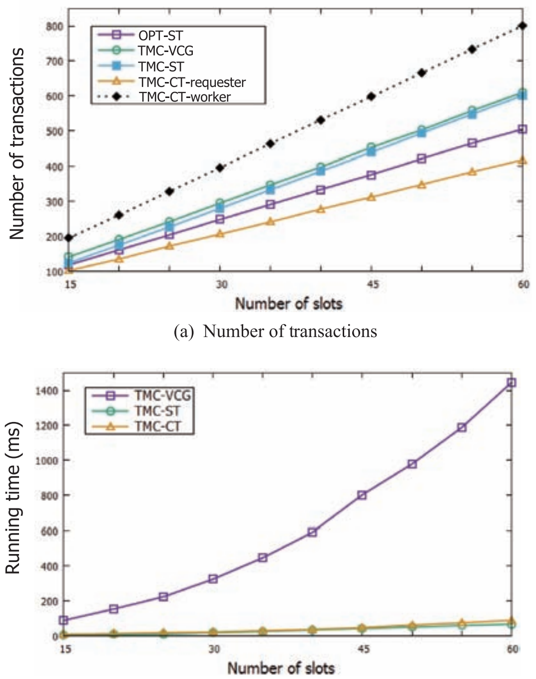
Fig. 6 Performance for complex task assignment.

different incentive mechanisms. Figure 7a shows that the number of transactions almost linearly increases as the number of slots increases. It shows that TMC-VCG has the highest number of transactions compared to TMC-ST and OPT-ST. TMC-ST distributes task greedily while with a lower number of transactions compared to TMC-VCG. Clearly, TMC-VCG has the advantages in the number of transactions. OPT-ST achieves the highest social welfare while sacrificing the number of deals. The requesters in TMC-CT demand multiple skills; therefore, it may be difficult to satisfy their demand. Hence, the requesters in TMC-CT gain the lowest turnover. However, the workers in TMC-CT achieve the highest deal rate because of the team formation between workers.

We conducted the running time test to confirm our time complexity analysis for the three mechanisms. Figure 7b clearly shows that TMC-VCG has the highest running time because of complex iterations, especially in the pricing stage. TMC-CT consuming most of the time in computing critical value has a higher running time compared to TMC-ST. Moreover, the growth rate of TMC-CT is higher than TMC-ST.

8 Conclusion

This study is inspired by the situation that self-interested players in crowdsourcing may improve utility by misreporting their bid or ask. We designed three truthful mechanisms for crowdsourcing, namely, TMC-VCG, TMC-ST, and TMC-CT. Among them, a traditional VCG-like auction scheme, TMC-VCG, is designed for simple task assignment, but not budget balance with a high time complexity. Therefore, we further propose TMC-ST for simple task assignment with a reachable time complexity. TMC-CT is conducted in the context of a complex task assignment, where the requester requires the collective efforts of multiple workers. We examine the truthfulness and other properties of each of the three mechanisms in the analysis section. In the performance section, we verified the truthfulness and analyzed the performance for simple task assignment and complex task assignment, respectively.



(a) Number of transactions

(b) Running time

Fig. 7 Comparison of mechanisms.

In the future, we will consider the social network of workers in crowdsourcing. The requester who posts a complex task such as software engineering would like to employ a team with close relationship. In other words, the workers prefer to cooperate with familiar teammates, so that the task completion rate can be improved accordingly.

Acknowledgment

This study was funded in part by the National Natural Science Foundation of China (Nos. 61472344 and 61611540347); Natural Science Foundation of Jiangsu Province (No. BK20150460); Six Talent Peaks Project in Jiangsu Province (No. 2011-DZXX-032); Scientific Research Foundation of Graduate School of Jiangsu Province (No. KYLX16_1391); and Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China (CAAC) (No. CAAC-ITRB-201604).

References

[1] J. Howe, Crowdsourcing: A definition, http://crowdsourcing: typepad:com/cs/2006/06/crowdsourcing a:html, Jun. 2006.

[2] A. Slivkins and J. W. Vaughan, Online decision making in crowdsourcing markets: Theoretical challenges, *ACM SIGecom Exchanges*, vol. 12, no. 2, pp. 4–23, 2013.

[3] Amazon Mechanical Turk, http://www.mturk.com/, 2017.

[4] TopCoder, https://www.topcoder.com/, 2017.

[5] A. I. Chittilappilly, L. Chen, and S. Amer-Yahia, A survey of general-purpose crowdsourcing techniques, *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2246–2266, 2016.

[6] U. U. Hassan and E. Curry, Efficient task assignment for spatial crowdsourcing, *Expert Systems with Applications: An International Journal*, vol. 58, no. C, pp. 36–56, 2016.

[7] D. Yang, X. Fang, and G. L. Xue, Truthful auction for cooperative communications, in *ACM International Symposium on Mobile Ad Hoc Networking and Computing*,

2011, pp. 89–98.

[8] W. J. Shi, L. Q. Zhang, C. Wu, Z. P. Li, and F. C. M. Lau, An online auction framework for dynamic resource provisioning in cloud computing, *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2060–2073, 2015.

[9] A. L. Jin, W. Song, P. Wang, D. Niyato, and P. J. Ju, Auction mechanisms toward efficient resource sharing for cloudlets in mobile cloud computing, *IEEE Transactions on Services Computing*, vol. 9, no. 6, pp. 895–909, 2015.

[10] S. Gujar and B. Faltings, Auction based mechanisms for dynamic task assignments in expert crowdsourcing, in *International Workshop on Agent-Mediated Electronic Commerce and Trading Agents Design and Analysis*, 2015.

[11] V. Krishna, Auction theory, *Academic Press*, vol. 7, no. 3, pp. 289–297, 2002.

[12] P. Cheng, X. Lian, L. Chen, J. S. Han, and J. Z. Zhao, Task assignment on multi-skill oriented spatial crowdsourcing, *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 8, pp. 2201–2215, 2016.

[13] S. Assadi, J. Hsu, and S. Jabbari, Online assignment of heterogeneous tasks in crowdsourcing markets, in *the Third AAAI Conference on Human Computation and Crowdsourcing*, 2015, pp. 12–21.

[14] T. Yue, S. Ali, and S. Wang, An evolutionary and automated virtual team making approach for crowdsourcing platforms, *Cloud-Based Software Crowdsourcing*, pp. 113–130, 2015.

[15] Q. Liu, T. Luo, R. M. Tang, and S. Bressan, An efficient and truthful pricing mechanism for team formation in crowdsourcing markets, in *2015 IEEE International Conference on Communications (ICC)*, 2015, pp. 567–572.

[16] X. Zhang, G. L. Xue, R. Z. Yu, D. J. Yang, and J. Tang, Truthful incentive mechanisms for crowdsourcing, in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2830–2838.

[17] Y. Fan, H. L. Sun, Y. M. Zhu, X. D. Liu, and J. Yuan, A truthful online auction for tempo-spatial crowdsourcing tasks, in *2015 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, 2015, pp. 332–338.

[18] A. Biswas, S. Jain, D. Mandal, and Y. Narahari, A truthful budget feasible multi-armed bandit mechanism for crowdsourcing time critical tasks, in *the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 2015, pp. 1101–1109.

[19] A. Subramanian, G. S. Kanth, and R. Vaze, Offline and online incentive mechanism design for smart-phone crowdsourcing, arXiv Preprint arXiv: 1310.1746, 2013.

[20] R. Myerson, Optimal auction design, *Mathematics of Operations Research*, vol. 6, no. 1, pp. 58–73, 1981.

[21] A. Gopinathan, Z. P. Li, and C. Wu, Strategyproof auctions for balancing social welfare and fairness in secondary spectrum markets, in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 3020–3028.

[22] Y. Gui, Z. Z. Zheng, F. Wu, X. F. Gao, G. H. Chen, SOAR: Strategy-proof auction mechanisms for distributed cloud bandwidth reservation, in *IEEE International Conference on Communication Systems*, 2014, pp. 162–166.

[23] T. Kameda and J. I. Munro, A $O(|V| * |E|)$ algorithm for maximum matching of graphs, *Computing*, vol.12, no. 1, pp. 91–98, 1974.

[24] L. Tran-Thanh, A. Chapman, A. Rogers, and N. R. Jennings, Knapsack based optimal policies for budget-limited multi-armed bandits, in *AAAI Conference on Artificial Intelligence*, 2012, pp. 1134–1140.

[25] L. Tran-Thanh, S. Stein, A. Rogers, and N. R. Jennings, Efficient crowdsourcing of unknown experts using multi-armed bandits, in *European Conference on Artificial Intelligence*, 2012, pp. 768–773.

[26] D. C. Parkes, J. Kalagnanam, and M. Eso, Achieving budget-balance with vickrey-based payment schemes in exchanges, in *17th International Joint Conference on Artificial Intelligence*, 2001, pp. 1161–1168.

[27] J. Edmonds and R. M. Karp, Theoretical improvements in algorithmic efficiency for network flow problems, *Journal of the ACM (JACM)*, vol. 19, no. 2, pp. 248–264, 1972.

Yonglong Zhang currently works at College of Information Engineering, Yangzhou University, China and is a PhD candidate at Nanjing University of Aeronautics and Astronautics, China. He received the MS degree in computer science from Nanchang University, China in 2004. His current research interests include mechanism design, auction, and cloud computing.

Haiyan Qin is a master student at College of Information Engineering, Yangzhou University, China. She received the bachelor degree from Yangzhou University, China in 2015. Her main research interests include game theory, auction, and task assignment in crowdsourcing.

Bin Li received the BS degree from Fudan University, China in 1986, MS and PhD degrees from Najing University of Aeronautics & Astronautics, China in 1993 and 2001, respectively. He is now a professor in Yangzhou University, China. He has published more than 100 journal and conference papers. His main research interests include artificial intelligence, multi-agent system, and service oriented computing.

Jin Wang received the BS and MS degrees from Nanjing University of Posts and Telecommunications, China in 2002 and 2005, respectively, and PhD degree from Kyung Hee University, Korea in 2010. Now, he is a professor in Nanjing University of Information Science and Technology. His research interests mainly include routing protocol and algorithm design and performance evaluation for wireless sensor networks.

Sungyoung Lee received the BS degree from Korea University, MS and PhD degrees in computer science from Illinois Institute of Technology (IIT) in 1987 and 1991, respectively. He has been a professor at Kyung Hee University since 1993. His current research focuses on ubiquitous computing and applications, wireless ad-hoc and sensor networks, context-aware middleware, sensor operating systems, real-time systems, and embedded systems.

Zhiqiu Huang received the PhD degree in computer Science from Nanjing University of Aeronautics and Astronautics, China in 1999. Now he is a professor and PhD supervisor at College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. His research interests include software engineering, formal methods, cloud computing, and privacy.