# Hiring a Team from Social Network: Incentive Mechanism Design for Two-tiered Social Mobile Crowdsourcing

Jia Xu, *Senior Member, IEEE,* Zhuangye Luo, Chengcheng Guan, Dejun Yang, *Senior Member, IEEE,* Linfeng Liu, *Member, IEEE,* Yan Zhang, *Fellow, IEEE*

**Abstract**—Mobile crowdsourcing has become an efficient paradigm for performing large scale tasks. The incentive mechanism is important for the mobile crowdsourcing system to stimulate participants, and to achieve good service quality. In this paper, we focus on solving the insufficient participation problem for the budget constrained online crowdsourcing system. We present a two-tiered social crowdsourcing architecture, which can enable the selected registered users to recruit their social neighbors by diffusing the tasks to their social circles. We present three system models for two-tiered social crowdsourcing system based on the arrival modes of registered users and social neighbors: offline model, semi-online model, and full-online model. We consider the tasks are associated with different end times. We present an incentive mechanism for each of three system models. Through both rigorous theoretical analysis and extensive simulations, we demonstrate that the proposed incentive mechanisms achieve computational efficiency, individual rationality, budget feasibility, cost truthfulness, and time truthfulness. We further show that our incentive mechanisms for semi-online model and full-online model can obtain averagely 51.1% and 39.7% value of approximate optimal untruthful offline algorithm, respectively.

**Index Terms**—Mobile crowdsourcing, incentive mechanism, social network, online mechanism.

## 1 Introduction

Nowadays, smartphones become almost indispensable to our lives. Smartphones are integrated with a variety of embedded sensors such as camera, light sensor, GPS, accelerometer, digital compass, gyroscope, microphone, and proximity sensor. These sensors can collectively monitor diverse human activities and the surrounding environment. Compared with the traditional sensor networks, mobile crowdsourcing has a huge potential due to the prominent advantages, such as wide spatio-temporal coverage, low cost, good scalability, and pervasive application scenario. It can be applied in various domains, such as Sensorly [2] for constructing cellular/WiFi network coverage maps, Nericell [3] and VTrak [4] for providing traffic information, as well as Ear-Phone [5] and NoiseTube [6] for creating noise maps.

There have been many research efforts on incentive mechanism design for mobile crowdsourcing [7, 8, 9]. Online incentive mechanism [10, 11] aims to deal with the mobile crowdsourcing, where the users arrive one by one in random order and user availability will change over time, and enables the decision on whether to buy users' service based on the current information.

However, most of the online mechanisms assume that there are enough participants in the mobile crowdsourcing systems. In reality, however, many tasks cannot be completed in time due to the insufficient participation. The bases of participants of crowdsourcing applications are still not big enough. According to [12], mobile crowdsensing applications have rarely scaled up to more than 1000 participants. According to the data of the fourth quarter in 2016 from Analysys [13], only 6.02% and 3.83% of all registered users can provide the real-time sensing data for the traffic condition in Tencent map and Tianyi navigation, respectively. Moreover, the crowdsourcing platform also benefits the developed platform when it cannot find enough workers interested in some specific tasks. The tasks requested by various crowdsourcers would require professional workers to complete. For example, an important proportion of Human Intelligence Tasks (HITs) in Amazon Mechanical Turk (AMT) [14] requires the workers to complete a test in order to be qualified. Our statistics data showed that there are 21.1 uncompleted requests that were publicized more than 2 weeks in AMT on average from 2021-5-19 to 2021-5-30, while each request may include several HITs (Human Intelligence Tasks). Among these requests, 79.3% requests were publicized more than one month. Another observation from Freelancer [15] from 2021-5-19 to 2021-5-30 showed that there were 51.9 uncompleted projects that were publicized more than 2 weeks on average. Among them, 60.1% projects were publicized more than one month. The above surveys reveal the insufficient participant problem of current crowdsourcing systems. In addition, at the

- *J. Xu, Z. Luo, C. Guan and L. Liu are with the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023, China. (e-mail: {xujia, 1019041114, 1216043121, liulf}@njupt.edu.cn).*
- *D. Yang. is with Colorado School of Mines, Golden, CO 80401. (e-mail: djyang@mines.edu).*
- *Y. Zhang is with Department of Informatics, University of Oslo, 0316 Oslo, Norway (e-mail:yanzhang@ieee.org).*
- *Corresponding author: L. Liu (e-mail:liulf@njupt.edu.cn).*

*This is an extended and enhanced version of the paper [1] that appeared in IEEE SECON 2018. This research was supported in part by NSFC grants 61872193, 61872191, 62072254 and 61772285, and NSF grants 1717315.*

early beginning, the crowdsourcing platform faces the cold-start problem and cannot provide sufficient workers for completing tasks. Thus how to expand the user pool of the crowdsourcing system is a nontrivial issue. As far as we known, there is no off-the-shelf online incentive mechanism designed in the literature for recruiting the users out of the crowdsourcing system to perform the tasks.

To address the insufficient participation problem, we extend the mobile crowdsourcing systems to the social networks in order to recruit more participants. We consider that the crowdsourcing platform is operated by an online community. Thus the platform can extract the personal profile of users in the online community. This assumption is reasonable and pervasive since many online communities have developed crowdsourcing systems themselves, such as Stepes [16] operated by Facebook, Google Image Labeler [17] and Translate Community [18] operated by Google+, QQ-Crowd [19] owned by QQ, Crowdtesting [20] and Baidu Baike [21] operated by Baidu.

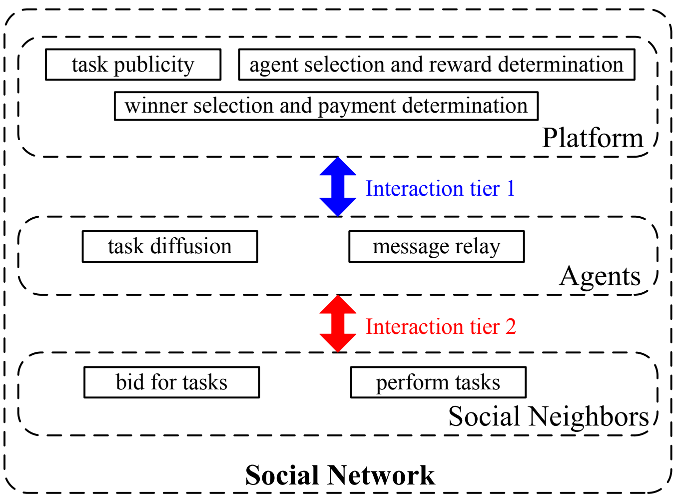



Fig. 1. Two-tiered social crowdsourcing system

In this paper, we propose a two-tiered social crowdsourcing architecture, which is illustrated by Fig. 1. In the proposed crowdsourcing system, a set of agents are selected from the registered users of the crowdsourcing system. The selected agents are in charge of diffusing the crowdsourcing tasks to their social neighbors through the social circle, such as Twitter, Microblog, Facebook, and WeChat. The social neighbors cannot interact with the platform directly since they haven't been registered with the crowdsourcing system. Thus, differently from most existing mobile crowdsourcing systems, there are two interaction tiers in the crowdsourcing system.

The online arrival of users is a more realistic setting for most crowdsourcing application systems since the users are not always ready all the time, and cannot wait the decision of task allocation for a long time [10, 11]. This is an especially important consideration for our two-tiered social crowdsourcing architecture because the task performers are social neighbors rather than the professional workers.

The incentive mechanism design for the registered users to perform tasks in mobile crowdsourcing system has been studied extensively [10, 11, 22]. Thus, we focus on addressing the insufficient participation problem in crowdsourcing, and only consider the incentives to the agents for diffusing tasks and the social neighbors for performing tasks.

To find the suitable agents, the influence of registered users should be calculated. Many topology-based social influence calculation methods have been proposed [23, 24, 25, 26]. However, these influence calculation methods are unsuitable for crowdsourcing context. First, the current online communities such as Twitter, Microblog, Facebook, and WeChat are usually large-scale. Therefore, the time complexity of computing the topology-based measures such as eigenvector centrality [23], degree centrality [24], betweenness centrality [25] and closeness centrality [26] over whole social network is very high. Moreover, measuring the influence of users only based on the network topology in crowdsourcing context is not sufficient because the users with good metrics of network structure probably cannot perform the tasks. Finally, it is hard to obtain the global knowledge of network structure in crowdsourcing system. Although many online communities have developed their own crowdsourcing systems, only limited knowledge can be obtained in most practical situations.

To address these issues, we only recruit the social neighbors of registered users of crowdsourcing platform. By this way, only the network structure of one-hop is needed, therefore, both the time complexity and the required topology knowledge can be largely reduced, improving the applicability and practicability of designed mechanisms. Note that once the social neighbors participate in the crowdsourcing tasks, they can be viewed as the new registered users, and the next round of social crowdsourcing can be launched with a larger registered user pool. Moreover, for influence calculation, we take into consideration the matching degree of task types and social neighbors' interests. In the context of crowdsourcing, the impact of tasks diffused is more important than the influence of agents to the social neighbors. Considering the purpose of recruiting participants to perform the crowdsourcing tasks, our method is more suitable for task diffusion, comparing with traditional influence computing only based on topology-based measures.

The problem of designing truthful incentive mechanism for the two-tiered social crowdsourcing system is very challenging. First, the new role of agent is introduced into the system, and a realistic system model, including the interactions between the agents and crowdsourcing platform and social neighbors, should be defined to enable the task diffusion through the two-tiered social crowdsourcing system. Second, since the mobile crowdsourcing system works online, the designed incentive mechanism should decide whether to accept the service or not, and at what price before the tasks expire and the social neighbors depart. It is challenging to satisfy the desirable properties of individual rationality, budget feasibility, and truthfulness simultaneously in online setting. Moreover, for our two-tiered social crowdsourcing system, both agents and social neighbors should be selected. It is a challenging problem to select appropriate agents to diffuse the tasks to the social neighbors who are interested in performing tasks to complete tasks as many as possible. In addition, to reduce the complexity of designed mechanism and the knowledge requirement of social network, only limited knowledge can be used to evaluate the influence of registered users. Finally, the social neighbors and the registered users may take a strategic behavior by submitting dishonest bid price and arrival/departure time to maximize their utilities.

The main contributions of this paper are as follows:

- We present the two-tiered social crowdsourcing architecture, which select and enable a set of agents to recruit the users from the social circle, to solve the insufficient participation problem as well as the cold-start problem of crowdsourcing systems.
- We propose three system models: offline model, semi-online model, and full-online model for the proposed social crowdsourcing architecture based on the availability of registered users and social neighbors.
- We design the incentive mechanism for each model. We show that the designed incentive mechanisms satisfy computational efficiency, individual rationality, budget feasibility, and truthfulness.
- The results of simulations based on the real-world data set show that our incentive mechanisms for semi-online model and full-online model can obtain averagely 51.1% and 39.7% value of approximate optimal untruthful offline algorithm, respectively.

The rest of the paper is organized as follows. We review the related work in Section 2. Section 3 formulates three system models and lists desirable properties. The designs of incentive mechanisms for three models are presented in Section 4, Section 5, and Section 6, respectively. Performance evaluation is shown in Section 7. We give the discussion in Section 8, and conclude this paper in Section 9.

## 2 Related Work

### 2.1 Offline Incentive Mechanisms

Many incentive mechanisms for mobile crowdsourcing have been proposed [27, 28, 29, 30]. Bhattacharjee *et al.* proposed an event-trust and user-reputation model, called *QnQ* model [31], where the user reputation scores are based on both quality (accuracy of contribution) and quantity (degree of participation) of their contributions, to segregate different user classes such as selfish, malicious and honest. Yang *et al.* proposed two models of smartphone crowdsourcing [32]: the platform-centric model where the platform provides a reward shared by participating users, and the user-centric model where users have more control over the payment they will receive. Jin *et al.* studied the *QoI* problem of crowdsensing, and designed the incentive mechanisms based on reverse combination auction for single-minded and multi-minded scenarios [33]. Barnwal *et al.* extended the PS-Sim framework with a novel budget allocation mechanism for incentivizing participants [34]. However, all the studies above mentioned do not aim at solving insufficient user participation problem. Moreover, they designed the incentive mechanisms only for offline case. Chen *et al.* proposed to incorporate sensing platform and social network applications, which already have large user bases, to build a three-layer network model. Furthermore, they designed the incentive mechanisms for both intermediaries and the crowdsensing platform [35]. However, they only considered the offline case without budget constraint.

### 2.2 Online Incentive Mechanisms

Online auction is the essence of many networked markets. The information about goods, agents, and outcomes is revealed one by one online in a random order, and the agents must make irrevocable decisions without knowing future information in online auction. Zhao *et al.* proposed *OMZ* and *OMG* models, which follow the multiple-stage sampling-accepting process [10]. At every stage, the mechanism allocates tasks to a smartphone user only if its marginal density is not less than a certain density threshold computed using previous users' information. However, Zhao's method assumes that both the arrival time of the users and the value from the users are equally distributed over the time. In this paper, the selected agents may have different influence to their social neighbors. The agent with high influence can recruit the social neighbors who are more interest in performing crowdsourcing tasks. Thus, *OMZ* and *OMG* cannot help to attract high valued users as well as to improve task completion level. Xiao *et al.* proposed an online task assignment for crowdsensing in predictable mobile social networks [11]. The *LOTA* algorithm for the *Minimum-Largest-Makespan* task assignment problem follows the greedy strategy, in which the requester assigns the task with the largest workload, in turn, to the earliest idle mobile user. Gao *et al.* formulated an optimization problem of maximizing the amount of high quality sensing data subject to the task budget, and proposed an effective and quality-aware online incentive mechanism to solve the problem [36]. However, [36] assumes that the platform is capable of recruiting enough users. In this paper, we not only take dynamic users into consideration, but also take dynamic tasks and agents into consideration.

### 2.3 Incentive Mechanisms with Social Network

Social networks have been extensively discussed in mobile crowdsourcing [37, 38, 39, 40]. When the participants are not enough to complete the sensing tasks, the social network can help to spread the tasks to more potential participants. Xu *et al.* studied the user compatibility problem for cooperative crowdsensing tasks in the online community, and designed truthful incentive mechanisms to minimize the social cost such that each of the cooperative tasks can be completed by a group of compatible users [37]. Stefano *et al.* defined a multi-layer social sensing framework to explore and quantify the dynamics patterns of interactions [38]. However, they aimed at evaluating and quantifying the role of homophily, network heterogeneity and multiplexity in the emergence and sustainability of cooperation on the social multiplex network of human users, and did not address the insufficient participation problem. Wang *et al.* proposed a game-theoretic team formation model by modeling each subtask as a cooperative mobile agent, and then each agent targets to move to the individual that has the least workload [41]. Nguyen *et al.* proposed the notions of node observability and coverage utility score, and designed a new context-aware approximation algorithm to find vertex cover that is tailored for crowd-sensing tasks in opportunistic mobile social networks [42]. The above research efforts utilize the social relationship to stimulate higher cooperative level, higher mobile participation level, or greater revenue of the crowdsensing service provider, and cannot recruit the social users out of the system and as well as solve the cold-start problem.

Han *et al.* formulated the participant recruitment in mobile opportunistic D2D networks from the perspective of the seeds, which face the dilemma of how to carefully invite additional participants in order to maximize their gain while keeping the risk of losing their payment low and propose a dynamic programming algorithm to solve it [43]. However, the problem is unique due to the opportunistic network setting. Luo *et al.* introduced a social concept called *nepotism* into participatory sensing to enhance the trustworthy and used Stackelberg game framework to maximize the utility of the sensing campaign organizer, while ensuring participants to strictly have incentive to participate [44]. However, they only considered the offline case, where all strategies can be calculated in advance.

Nie *et al.* modeled the interaction between the crowdsensing service provider and mobile users as a two-stage Stackelberg game, and investigated two types of incentive mechanism for the crowdsensing platform with complete and incomplete information on social network effects [45, 46]. They further proposed a multi-leader and multi-follower Stackelberg game approach to model the strategic interactions among multiple service providers and users [47]. Different from [47], our paper takes advantage of social influence to promote the crowdsourcing task diffusion to social users. Yang *et al.* proposed a framework for designing the social incentive mechanism to promote cooperation in crowdsensing, where the utility of a user largely relies on the participation of its social friends [48]. These studies aim at weighing the influence and its relationship among social users using game-theoretical approaches. However, weighing the influence of neighbors and the relationship with the choices of users require the social ties of whole network, which is usually hard to obtain. Moreover, in our online scenario, when making the choice, the user cannot know whether the social friends will be online and participate in the crowdsourcing. Thus, it is unlikely to calculate the relationship of users' choices exactly, i.e., it is hard to calculate the impact of social friends on his utility, therefore, the game-theoretical approaches would be ineffective. Take all these into account, we consider that the social neighbors make their decision (bid for tasks) independently, and their utilities are not affected by other social users' choices.

Overall, there is no off-the-shelf incentive mechanism designed in the literature, which recruits users from social network to perform the tasks in the online manner.

## 3 System Model and Desirable Properties

In this section, we present three different mobile crowdsourcing system models based on two-tiered social crowdsourcing architecture: offline model, semi-online model, and full-online model. At the end of this section, we present some desirable properties.

### 3.1 Offline Model of Two-tiered Social Crowdsourcing

In the offline model, both registered users and social neighbors are always ready to participate in the crowdsourcing. This means the registered users can diffuse the tasks immediately once the platform publicizes them, and the social neighbors can bid via the registered users for performing tasks in a short time. This is possible only in some special cases where the registered users have a large group of social neighbors and have great influence over their cycles. The main reason for studying this model is to provide a performance benchmark for realistic online models. To enable the benchmark to output an upper bound of performance, we consider the registered users are not profit-driven, i.e., the registered users can diffuse the tasks before they are selected as agents.

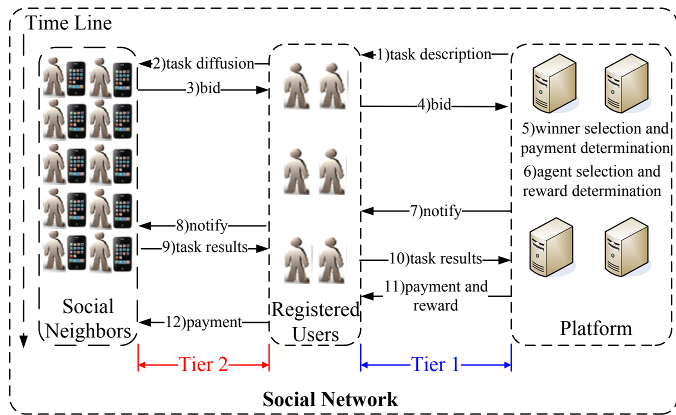



Fig. 2. Offline two-tiered social crowdsourcing system

As shown in Fig. 2, we consider that the platform publicizes a set of tasks $\Gamma$=$\{\tau_1, \tau_2, ..., \tau_m\}$ to registered users with budget, where each task $\tau_k \in \Gamma, k = 1, 2, ..., m$, is associated with an end time $e_k$ and a type $t_k$. The task types, such as translation, data collection, and image recognition, are predefined by the platform, and different tasks can have the same task type. For the sake of brevity, we consider all tasks are launched at time step one.

Assume that a set of registered users $J = \{1, 2, \ldots, n\}$ of the platform are interested in diffusing crowdsourcing tasks. There is a budget $B$, which is the maximum value that the platform is willing to pay for the participants when they complete the tasks. There is also a total reward $R$ for incentivizing the agents to diffuse the tasks.

Upon receiving the tasks, the registered users diffuse the tasks to their social cycles immediately. Let $SN^j$ represent the set of influenced social neighbors of any registered user $j \in J$, and $SN$ represent the set of influenced social neighbors of all registered users, i.e., $SN = \cup_{j\in J} SN^j$.

The influenced social neighbors can participate in the mobile crowdsourcing through a reverse auction, and each social neighbor can get the payment for providing the crowdsourcing service. In the reverse auction, each social neighbor $i \in SN^j$ submits a bid $\theta_i = (\Gamma_i, b_i, j)$ to the platform via registered user $j$, where $\Gamma_i \subseteq \Gamma$ is the task set he/she is willing to perform, and $b_i$ is the reserve price.

The platform calculates the winner set $S \subseteq SN$, and determines the payment $p_i$ for each social neighbor $i \in SN$. Meanwhile, the platform selects a subset of registered users $A \subseteq J$ as the agents, and determines the reward $r^j$ to each registered user $j \in J$ . Then the platform notifies the agents and the winning social neighbors of the determination. The winners submit the task results to the platform via the corresponding agents. Finally, each winner/agent obtains the payment/reward, which is determined by the platform.

We consider the real cost $c_i$ for performing $\Gamma_i$ is private and only known to social neighbor $i$. Since we consider the social neighbors are selfish and rational individuals, each social neighbor can behave strategically by submitting the dishonest reserve price to maximize its utility. Note that a social neighbor cannot lie about the task set it is willing to perform and the agent it is associated with since they can be directly verified by the platform or the agents.

We define the utility of social neighbor $i$ as the difference between the payment and its real cost:

$$u_i = p_i - c_i \tag{1}$$

Specifically, the utility of the losers would be zero because they are paid nothing in our designed mechanisms and there is no cost for performing tasks.

Given the task set $\Gamma$ and the bid profile $\mathbf{\Theta} = \{\theta_1, \theta_2 ..., \theta_{|SN|}\}$, budget $B$, and total reward $R$, the incentive mechanism $M(\Gamma, \mathbf{\Theta}, B, R)$ outputs the winner set $S \subseteq SN$, the payment vector $\mathbf{p} = (p_1, p_2, ..., p_{|SN|})$, the agent set $A \subseteq J$, and the reward vector $\mathbf{r} = (r^1, r^2, ..., r^n)$.

Let $v_i$ be the value of any winner $i$. Let $V(S)$ be the value function of the platform over the winner set $S$. The objective of our incentive mechanism is maximizing value from the winners' services under the budget constraint $B$ and the reward constraint $R$, i.e.

$$\max V(S) \quad s.t. \sum_{i \in S} p_i \leq B, \sum_{j \in A} r^j \leq R \tag{2}$$

In this study, we consider the value function $V(S)$ is nonnegative, monotone, and submodular, which is defined in Definition 1. Many value functions of crowdsourcing systems satisfy the submodularity [5, 8, 10, 49], which covers many realistic scenarios, such as [3, 36]. Note that the incentive mechanisms designed in this paper is effective for all nonnegative, monotone, and submodular value functions. The exact form value function is obviously subject to applications. For convenience, we adopt a simple linear cumulative function in our experiments: $V(S) = \sum_{i \in S} v_i = \sum_{i \in S} \sum_{\tau_k \in \Gamma_i} v(k)$, where $v_i$ is the value of social neighbor $i$, $v(k)$ is the value of task $\tau_k \in \Gamma$.

**Definition 1** *(Nonnegative Monotone Submodular Function): Let $\Omega$ be a finite set. A function $f$:$2^\Omega \to \mathbb{R}$ is submodular if and only if*

$$(1) f(X) \geq 0, for\ any X \subseteq \Omega$$
$$(2) f(X) \leq f(Y), for\ any X \subseteq Y \subseteq \Omega$$
$$(3) f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y), for\ any\ X \subseteq Y \subseteq \Omega, and\ x \in \Omega \setminus Y,$$

*where $2^\Omega$ is the power set of $\Omega$, $\mathbb{R}$ is the set of reals.*

### 3.2 Semi-online Model of Two-tiered Social Crowdsourcing

In semi-online model, we consider the registered users are always ready to participate in crowdsourcing. The professional workers are employed in crowdsourcing systems, such as the full-time photo annotator in Google Image Labeler [17], and the full-time software testers in QQ-Crowd [19] and Crowdtesting [20]. The registered users even can develop the automatic bidding program, which is available all the time, to bid for task diffusion automatically. However, the social neighbors do not keep in touch with the registered users all the time since they are not the professional workers of crowdsourcing. The social neighbors arrive one by one in a random order, and user availability changes over time.

There are two key differences from the offline model. First, to make our model more practical, we consider the registered users are profit-driven, i.e., the registered users only diffuse the tasks after they are selected as agents. This means the platform needs to select the agents before selecting winners in order to diffuse tasks. Second, each social neighbor has an arrival time and a departure time, and is only available before it departs.

Since the registered users are always online, we model the mobile crowdsourcing in this case as a dual budget feasible auction, which enables our incentive mechanism to work effectively under the actual situation with rational registered users. We first select the agents through a budget feasible reverse auction. Then we select the winners through an online budget feasible reverse auction.

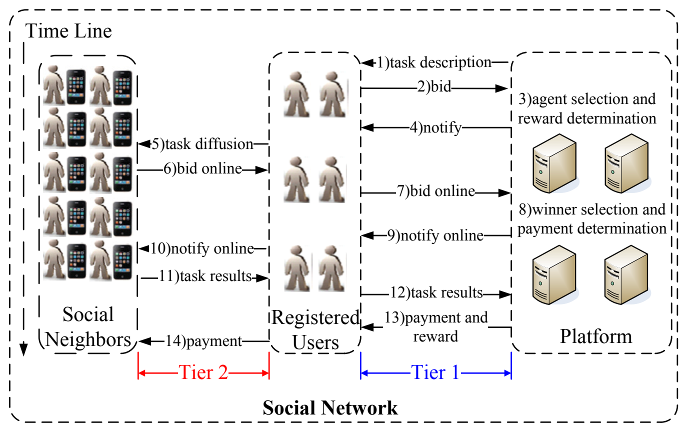



Fig. 3. Semi-online two-tiered social crowdsourcing system

Now we give the workflow of mobile crowdsourcing in the semi-online model, and highlight the differences between the offline model and the semi-online model for the sake of brevity. The platform publicizes a set of tasks to registered users. Each registered user $j \in J$ bids for task diffusion with reserve price $b^j$. The platform selects the agents from the bidders, and determines the reward for all registered users. Once the registered users are selected as the agents, they diffuse the tasks to their social neighbors. Each social neighbor $i \in SN^j$ submits a bid $\theta_i = (a_i, d_i, \Gamma_i, b_i, j)$ to the platform via agent $j$ when it is online, where $a_i$ and $d_i$ are the arrival time and departure time of social neighbor $i$, respectively. When any bid $\theta_i$ is submitted to the platform via an agent $j$, the platform needs to decide whether to buy the service of social neighbor $i$, and if so, at what price $p_i$ before $i$ departs. Then the platform notifies the winners via agents. The winners perform the tasks and submit the results to the platform via the agents before their departure. Finally, each winner/agent obtains the payment/reward. The whole process is illustrated by Fig. 3.

We consider the real cost $c_i$ for performing $\Gamma_i$, the real arrival time $ra_i$, and departure time $rd_i$ are private information. Each social neighbor can behave strategically by submitting the dishonest reserve price, or dishonest arrival/departure time to maximize its utility. Note that a social neighbor cannot announce an earlier arrival time or a later departure time than its true arrival/departure time, i.e., $ra_i \leq a_i \leq d_i \leq rd_i$. This is justified since the presence can be directly verified by the platform. We also consider that each registered user can behave strategically by submitting the dishonest reserve price to maximize its utility.

### 3.3 Full-online Model of Two-tiered Social Crowdsourcing

In this subsection, we consider the full-online model, which is closer to the real-world mobile crowdsourcing. Differently from the semi-online model, we consider that the registered users are not always ready to participate in crowdsourcing. This is a pervasive observation because most contributors of crowdsourcing are not professional workers, and have their own available time.

Since the registered users are profit-driven, the platform needs to select the agents before selecting winners. However, the registered users arrive at the platform in an asynchronous way. We cannot apply any offline auction to select the agents. To address this issue, we present the workflow of mobile crowdsourcing in the full-online model, which is illustrated by Fig. 4. Again, we focus on the differences between the semi-online model and the full-online model.

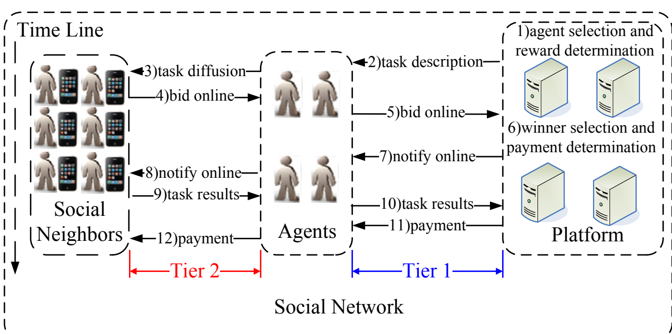



Fig. 4. Full-online two-tiered social crowdsourcing system

The platform first selects at most $\gamma$ agents from all registered users and determines the reward to the agents, where $\gamma$ depends on the total reward $R$ and the mean of the diffusion cost. The diffusion cost can be estimated through the history or making a customer survey. There are many public sources that can help to estimate the cost [50, 51].

Then the platform publicizes the tasks. We consider the agents arrive at the platform in an asynchronous way. Each agent $j \in A$ has an arrival time $a^j$ and a departure time $d^j$, $max\{e_1, e_2 ..., e_m\} \geq d^j \geq a^j \geq 0$. Note that the agents cannot state an early arrival time or late departure time in practice. Once any agent $j$ is online, it will send a message $MSG = (\Gamma^j, a^j, d^j)$, where $\Gamma^j$ is the set of unexpired tasks when $j$ arrives, to the platform. After confirmed by the platform, the agent $j$ sends the same message to its social circle in order to diffuse the tasks.

TABLE 1
Frequently used notations

| Notation | Description |
|---|---|
| $\Gamma, \tau_k, e_k$ | task set, task $k$, end time of task $\tau_k$ |
| $m, n$ | number of tasks, number of registered users |
| $J, A, S$ | registered user set, agent set, winner set |
| $B, B', R$ | budget, stage budget, total reward |
| $\mathscr{B}^j, \boldsymbol{\mathscr{B}}$ | budget of agent $j$, budget profile of agents |
| $SN, SN^j$ | social neighbor set, social neighbor set of registered user $j$ |
| $\Theta, \theta_i$ | bid profile of social neighbors, bid of social neighbor $i$ |
| $b_i, c_i$ | reserve price of social neighbor $i$, cost of social neighbor $i$ |
| $\mathbf{b}, b^j$ | bid profile of registered users, reserve price of registered user $j$ |
| $\Gamma_i$ | task set of social neighbor $i$ |
| $\Gamma^j$ | set of unexpired tasks when agent $j$ arrives |
| $a_i, d_i$ | arrival time and departure time of social neighbor $i$ |
| $a^j, d^j$ | arrival time and departure time of agent $j$ |
| $ra_i, rd_i$ | real arrival time and real departure time of social neighbor $i$ |
| $\mathbf{p}, p_i$ | payment vector, payment to social neighbor $i$ |
| $\mathbf{r}, r^j$ | reward vector, reward to agnet $j$ |
| $V(S)$ | value to the platform over winner set S |
| $v_i, v^j$ | value of social neighbor $i$, value of agent $j$ |
| $u_i$ | utility of social neighbor $i$ |
| $Jac(\cdot)$ | jaccard similarity coefficient |
| $I(\cdot), I_{max}$ | influence function, maximum influence |
| $Inf_i^j$ | influence of registered user $j$ to social neighbor $i \in SN^j$ |
| $Inf^j, Inf(A)$ | influence of registered user $j$, influence over agent set $A$ |
| $T, T', \boldsymbol{t}$ | deadline, end time step of each stage, time step |
| $\mathscr{O}, \mathscr{O}'$ | online social neighbor set, unselected online social neighbor set |
| $\delta$ | unit influence threshold |
| $\gamma$ | maximum number of agents selected in full-online model |

Each social neighbor $i \in SN^j$ submits a bid $\theta_i = (a_i, d_i, \Gamma_i, b_i, j)$ to the platform via agent $j$ when it is online. We consider $d^j \geq d_i \geq a_i \geq a^j$ since a social neighbor needs to submit its bid and receive the notice of determination via agent. The platform selects the winners and determines the payment to the social neighbors. Then the platform notifies the winners via agents. The winners perform the tasks and submit the results to the platform via the agents before their departure. Finally, each winner obtains the payment.

We list the frequently used notations in TABLE 1.

### 3.4 Desirable Properties

Our objective is to design the incentive mechanisms satisfying the following four desirable properties:

- **Computational efficiency:** An incentive mechanism is computationally efficient if the agent set $A$, the winner set $S$, the reward $\mathbf{r}$, and the payment $\mathbf{p}$ can be computed in polynomial time.
- **Individual Rationality:** Each social neighbor will have a non-negative utility while reporting the true cost, and the true arrival/departure time, i.e., $u_i \geq 0, \forall i \in SN$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3162108, IEEE Transactions on Mobile Computing



- **Budget Feasibility:** The mechanism is budget feasible if the total payment to the social neighbors is smaller or equal to the budget $B$, i.e., $\sum_{i\in S} p_i \leq B$.
- **Truthfulness:** A mechanism is cost truthful and time truthful (or simply called truthful) if no social neighbor can improve its utility by submitting false cost or false arrival/departure time, no matter what others submit. We say the incentive mechanism is universally truthful if it takes a random distribution over deterministic truthful incentive mechanisms.
- **Approximation:** The objective function is maximizing the value over the winner set S. We attempt to find optimal solution or approximation algorithm with low approximation ratio when there is no optimal solution computed in polynomial time. For the latter, the approximation ratio, $O(g(n))$, is the maximum ratio between optimal solution and approximation solution. Specifically, we call the solution is with constant approximation ratio if the maximum ratio between the optimal solution and the approximation solution is within a constant.

The importance of the first three properties is obvious because they together assure the feasibility of the incentive mechanism. Truthfulness is indispensable for guaranteeing the compatibility. Being truthful, the incentive mechanism can eliminate the fear of market manipulation and the overhead of strategizing over others for the users. The last property measures the performance of incentive mechanism.

## 4 Incentive Mechanism under Offline Model

In this section, we present an *Incentive Mechanism for the Two-tiered Social Crowdsourcing System under Offline Model (MTSO)*. Considering the desirable property of truthfulness, our problem falls into the research on *Budget Feasible Submodular Maximization Mechanism Design*, which has been extensively studied [52, 53, 54].

The basic idea of *MTSO* is performing a budget feasible offline auction after collecting the bids from social neighbors. Rather than the deterministic algorithm [52] using a single criterion with approximation ratio of 112, we apply the random mechanism proposed by Chen [53], which has been proved to achieve properties of individual rationality, budget feasibility, truthfulness, and 5-approximation of the optimum [54], to select winners and determine the payment. The random mechanism selects the social neighbor who has the maximum value with reserve price no more than the budget as the winner with probability 2/5. With probability 3/5, the random mechanism selects the social neighbor according to the ratio of marginal value to the reserve price iteratively until its reserve price is large enough. By setting the specific probabilities, we will show that the random algorithm is 5-approximation. Then, we select all registered users with winning social neighbors as the agents, and allocate the reward based on the contribution to the value obtained by the platform.

The whole process is illustrated in Algorithm 1. Let $S^*$ be the set of social neighbors whose reserve price is no more than the budget (Line 1). With probability $\frac{2}{5}$ (Lines 7-9), we select the social neighbor $i^*$ with maximum value in set $S^*$ as the winner, and the payment is equal to the budget. The corresponding registered user $j_{i^*}$ of $i^*$ is select as the agent. We allocate the total reward to $j_{i^*}$. With probability $\frac{3}{5}$ (Lines 9-29), MTSO consists of winner selection and agent selection phase (Lines 10-16), payment determination phase (Lines 17-25), and reward selection phase (Lines 26-28).

**Algorithm 1 : MTSO**

**Input:** task set $\Gamma$, bid profile $\Theta = (\theta_1, \theta_2, ..., \theta_{|SN|})$, budget $B$, total reward $R$

1: $(S, A, \mathbf{p}, \mathbf{r}) \leftarrow (\emptyset, \emptyset, 0, 0)$; $S^* \leftarrow \{i | b_i \leq B\}$;
2: **for** each $j \in J$ **do**
3: $\quad v^j = 0$;
4: **end for**
5: $i^* \leftarrow arg\ max_{i\in S^*} v_i$;
6: $\zeta \leftarrow random[0, 5]$;
7: **if** $\zeta < 2$ **then**
8: $\quad S \leftarrow \{i^*\}; p_i \leftarrow B; A \leftarrow \{j_{i^*}\}; r^{j_{i^*}} \leftarrow R$;
9: **else**
//winner selection
10: $\quad i \leftarrow arg\ max_{i'\in S^*} \frac{V_{i'}(S)}{b_{i'}}$;
11: $\quad$ **while** $b_i \leq \frac{B * V_i(S)}{2V(S\cup\{i\})}$ **do**
12: $\quad\quad v^{j_i} \leftarrow v^{j_i} + V_i(S)$;
13: $\quad\quad S \leftarrow S \cup \{i\}$;
14: $\quad\quad A \leftarrow A \cup \{j_i\}$; //agent selection
15: $\quad\quad i \leftarrow arg\ max_{i'\in S^*\setminus S} \frac{V_{i'}(S)}{b_{i'}}$;
16: $\quad$ **end while**
//payment determination
17: $\quad$ **for** each $i \in S$ **do**
18: $\quad\quad S^{*'} \leftarrow S^* \setminus \{i\}$; $S' \leftarrow \emptyset$;
19: $\quad\quad i' \leftarrow arg\ max_{i''\in S^{*'}} \frac{V_{i''}(S')}{b_{i''}}$;
20: $\quad\quad$ **while** $b_i \leq \frac{B\times V_{i'}(S')}{2V(S'\cup\{i'\})}$ **do**
21: $\quad\quad\quad i' \leftarrow arg\ max_{i''\in S^{*'}\setminus S'} \frac{V_{i''}(S')}{b_{i''}}$;
22: $\quad\quad\quad p_i \leftarrow \max\left\{p_i, min\left\{\frac{B\times V_i(S')}{2V(S'\cup\{i\})}, \frac{V_i(S')\times b_{i'}}{V_{i'}(S')}\right\}\right\}$;
23: $\quad\quad\quad S' \leftarrow S' \cup \{i'\}$;
24: $\quad\quad$ **end while**
25: $\quad$ **end for**
//reward determination
26: $\quad$ **for** each $j \in A$ **do**
27: $\quad\quad r^j \leftarrow R\frac{v^j}{\sum_{j'\in A} v^{j'}}$;
28: $\quad$ **end for**
29: **end if**
30: **return** $(S, A, \mathbf{p}, \mathbf{r})$;

In winner selection and agent selection phase, we process each social neighbor $i \in S^* \setminus S$ iteratively according its marginal density $\frac{V_i(S)}{b_i}$, where $V_i(S)$ is the marginal value over set $S$ of selected winners, i.e., $V_i(S) = V(S \cup \{i\}) - V(S)$ (Lines 10). In each iteration, to achieve the budget feasibility, if the reserve price is no more than $\frac{B\times V_i(S)}{2V(S\cup\{i\})}$, the social neighbor $i$ is included in the winner set, and the corresponding registered user $j_i$ is included in the agent set (Lines 11-16). We set any agent $j$'s value $v^j$ as the total marginal value of $j$'s winning social neighbors (Line 12).

In payment determination phase, the payment to each winner should be determined to achieve the properties of individual rationality and truthfulness. For each winner $i \in S$, we execute the winner selection phase over $S^* \setminus \{i\}$, and denote the winner set as $S'$ (Lines 20-24). We apply the *modified proportional share allocation rule* [52] to achieve the critical value of payment. The payment to any winner $i$ is

$$p_i = max_{i' \in S'} \left\{ min \left\{ \frac{B \times V_i(S'_{i'-1})}{2V(S'_{i'-1} \cup \{i\})}, \frac{V_i(S'_{i'-1}) \times b_{i'}}{V_{i'}(S'_{i'-1})} \right\} \right\} \tag{3}$$

where $S'_{i'}$ is the winner set before we include $i'$ into $S'$.

In reward determination phase (lines 26-28), we allocate the reward according to the value proportion of each agent.

**Lemma 1**. *MTSO is computationally efficient.*

*Proof:* It suffices to analysis the time complexity of the second branch (Lines 9-29) of random mechanism since it dominates the running time of *MTSO*. Finding the user with maximum marginal density takes $O(|SN|m)$ time, where computing the value of $V_i(S)$ takes $O(m)$ time. Since there are $m$ tasks and each winner should contribute at least one new task to be selected, the number of winners is at most $m$. Hence, the while-loop (Lines 11-16) thus takes $O(|SN|m^2)$ time. In each iteration of the for-loop (Lines 17-25), a process similar to Lines 11-16 is executed. Hence the payment determination takes $O(|SN|m^3)$. The reward determination (Lines 26-28) takes $O(n)$ time. Since $n < |SN|$, the running time of *MTSO* is dominated by the payment determination phase, which is bounded by $O(|SN|m^3)$. □

We futher have the following theorem.

**Theorem 1**. *MTSO is computationally efficient, individually rational, budget feasible, universally truthful, and has approximation ratio of 5.*

*Proof:* The computational efficiency is analyzed in Lemma 1. The individual rationality, budget feasibility and universal truthfulness have been proved by [52]. next, we show that *MTSO* has approximation ratio of 5. Let $S_{opt}$ be the optimum. Let $l-1$ be the last social neighbor selected by Lines 11-16 of Algorithm 1, and the corresponding winner set is denoted as $S_{l-1}$. Based on the monotonicity and submodularity of value function, we have:

$$\begin{aligned} &v\left(S_{opt}\right) - v\left(S_{l-1}\right) \leq v\left(S_{opt} \cup S_{l-1}\right) - v\left(S_{l-1}\right) \\ &\leq \sum_{i \in S_{opt} \setminus S_{l-1}} V_i\left(S_{l-1}\right) = \sum_{i \in S_{opt} \setminus S_{l-1}} \frac{V_i\left(S_{l-1}\right)}{b_i} b_i \end{aligned} \tag{4}$$

Due to the fact of $l = \arg max_{i \in SN \setminus S_{l-1}} \frac{V_i(S_{l-1})}{b_i}$, we have $\frac{V_i(S_{l-1})}{b_i} \leq \frac{V_l(S_{l-1})}{b_l}$. Since $l$ is not a winner, we have $\frac{b_k}{V_l(S_{l-1})} > \frac{B}{2v(S_l)}$. Thus:

$$\begin{aligned} \sum_{i \in S_{opt} \setminus S_{l-1}} \frac{V_i\left(S_{l-1}\right)}{b_i} b_i &\leq \sum_{i \in S_{opt} \setminus S_{l-1}} b_i \frac{V_l\left(S_{l-1}\right)}{b_l} \\ &< B * 2 * \frac{v\left(S_l\right)}{B} = 2v\left(S_l\right) \end{aligned} \tag{5}$$

By using submodularity, we have:

$$2v\left(S_l\right) \leq 2\left(v\left(S_{l-1}\right) + v\left(l\right)\right) \leq 2\left(v\left(S_{l-1}\right) + v\left(i^*\right)\right) \tag{6}$$

By putting all the above together, we have:

$$v\left(S_{opt}\right) \leq 3v\left(S_{l-1}\right) + 2v\left(i^*\right) \tag{7}$$

Based on the Algorithm 1, the mechanism chooses $S_{l-1}$ with probability 3/5 and $i^*$ with probability 2/5. So we have:

$$5E\left(v\left(S\right)\right) = 3v\left(S_{l-1}\right) + 2v\left(i^*\right) \geq v\left(S_{opt}\right) \tag{8}$$

□

# 5 Incentive Mechanism under Semi-online Model

In this section, we present an *Incentive Mechanism for the Two-tiered Social Crowdsourcing System under Semi-online Model (MTSS)*. In this case, we execute the offline budget feasible auction and online budget feasible auction for selecting agents and winners, respectively.

## 5.1 Offline Budget Feasible Auction for Agent Selection

Since all social neighbors arrive in a random order, we cannot select the agents according to the value of their social neighbors. However, we consider that the crowdsourcing platform rides on an online community, the social neighbor of each registered user can be extracted from the personal profile of online community.

We expect to select the agents with social neighbors who are interested in performing tasks. The influence of any registered user depends on the matching degree of task types and its social neighbors' interests. Let $Inf(A)$ be the influence of all agents in $A$. The objective of our offline budget feasible auction is maximizing the total influence of agents under the reward constraint $R$, i.e.,

$$\max Inf(A) \tag{9}$$

$$s.t. \sum_{j \in A} r^j \leq R \tag{10}$$

Next, we characterize the influence of registered users. The ideal situation is that the types of tasks and the interests of social neighbor are exactly same. In this case, the social neighbor can complete all tasks theoretically, and the probability of completing all tasks is high. Otherwise, if the interests of social neighbor cannot cover the types of tasks, then the social neighbor must not complete all tasks. If the types of tasks cannot cover the interests of social neighbor, the social neighbor would not perform some of tasks, and the social neighbor probably cannot complete all tasks as well. So, we measure the similarity as the proportion of common interests (types) to all possible interests (types), i.e., *Jaccard Similarity Coefficient* [55], which is widely used to measure the set similarity [56]. Other set similarity measurements such as *Dice Coefficient*, *Simpson Coefficient* and *Ochiai Coefficient* cannot capture this characteristic well.

We use the *Jaccard Similarity Coefficient* $Jac(\Gamma, i)$ to measure how well the types of $\Gamma$ match the interests of any social neighbor $i \in SN$:

$$Jac(\Gamma, i) = \frac{|Type(\Gamma) \cap Interest(i)|}{|Type(\Gamma) \cup Interest(i)|} \tag{11}$$

where $Type(\Gamma) = \{t_k | \tau_k \in \Gamma, k = 1, 2, ..., m\}$ is the set of types of tasks in $\Gamma$, $Interest(i)$ is the interests of social neighbor $i$, which can be extracted from the personal profile of the online community.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3162108, IEEE Transactions on Mobile Computing



Considering the social neighbor's diminishing return on the influence of registered user, we introduce the influence function originated from task influence maximization [46] to measure the increase of influencing probability:

$$I(Z, I_{max}) = (I_{max} - 1)\sqrt{1-(1-Z)^2} + 1 \tag{12}$$

where $Z$ is the input probability and $I_{max}$ is the maximum influence, $I_{max} > 1$. Then we have $I(0, I_{max}) = 1, I(1, I_{max}) = I_{max}, \frac{\partial I(Z, I_{max})}{\partial Z} > 0$ and $\frac{\partial^2 I(Z, I_{max})}{\partial Z^2} < 0$ for $Z \in (0, 1)$.

We set the input probability as the *Jaccard Similarity Coefficient* given by (9), then the influence of any registered user $j$ to any social neighbor $i \in SN^j$ is defined as

$$Inf_i^j = I(Jac(\Gamma, i), I_{max}), i \in SN^j \tag{13}$$

We use $Inf_i^j$ to measure the possibility of $i$ bidding for performing any task in $\Gamma$ when any registered user $j \in J$ diffuses the task set $\Gamma$ to any social neighbor $i \in SN^j$. Then the influence of any registered user $j \in J$ is calculated as

$$Inf^j = \sum_{i \in SN^j} Inf_i^j \tag{14}$$

The influence calculated by (11)-(14) represents the ability of registered user to influence its social neighbors such that they will participate in the crowdsourcing, i.e., bidding for tasks. The calculated influence is irrelevant to either the number of publicized tasks or the number of tasks the social neighbors can perform. In other words, we aim to maximize the number of influenced and interested social neighbors rather than the task accomplish in agent selection stage. This is because it is difficult to estimate which and how many tasks the social neighbors can perform exactly. Therefore, as shown in semi-online model, we model the winner selection as an online budget feasible reverse auction, where the task subset of the bidders is determined by the submitted bids.

For the agent set $A$, the influence of all agents in $A$ can be calculated as

$$Inf(A) = \sum_{j \in A} Inf^j \tag{15}$$

Since $Inf(A)$ is an additive function, the above problem is the budget feasible mechanism design problem for maximizing additive valuations essentially. We introduce the random mechanism with approximation ratio of 2 (the best approximation ratio we know), proposed by Gravin [57]. The random mechanism selects the agents with large ratio of influence to the reserve price, and a random reward mechanism is employed. Illustrated in Algorithm 2, our reverse auction consists of agent selection phase and reward determination phase.

In the agent selection phase, we initialize the density threshold $\beta = \frac{1}{R} max_{j \in J} Inf^j$ (Line 9). For each registered user $j \in J$, we put it into the candidate set $S(\beta)$ if its unit influence $\frac{Inf^j}{b^j}$ is no less than the density threshold (Line 9). Then we continuously increase the value of density threshold (Lines 10-16) until $\beta \times R \geq Inf(S(\beta)) - max_{j \in S(\beta)} Inf^j$. In each iteration, if any candidate's unit influence is no more than the density threshold, we remove it from the candidate set (Lines 13-15). By this way, each registered user in candidate set is with the unit influence, which is more than the density threshold, and we put it into final agent set $A$ (Line 17).

**Algorithm 2 : Agent Selection of MTSS**

**Input:** bid profile $\mathbf{b} = (b^1, b^2, ..., b^n)$, total reward $R$, registered users $J$, task set $\Gamma$

1: $A \leftarrow \emptyset; \mathbf{r} \leftarrow 0$;
//agent selection
2: **for** each $j \in J$ **do**
3: $\quad Inf^j \leftarrow 0$;
4: $\quad$ **for** each $i \in SN^j$ **do**
5: $\quad\quad$ Calculate $Inf_i^j$ based on formula (13);
6: $\quad\quad Inf^j \leftarrow Inf_i^j + Inf^j$;
7: $\quad$ **end for**
8: **end for**
9: $\beta = \frac{1}{R} max_{j \in J} Inf^j; S(\beta) \leftarrow \{j | j \in J, \frac{Inf^j}{b^j} \geq \beta\}$;
10: **while** $\beta * R < Inf(S(\beta)) - max_{j \in S(\beta))} Inf^j$ **do**
11: $\quad k \leftarrow arg\ min_{j \in S(\beta)} \frac{Inf^j}{b^j}$;
12: $\quad \beta \leftarrow min\{\frac{Inf(S(\beta)) - max_{j \in S(\beta))} Inf^j}{R}, \frac{Inf^k}{b^k}\}$;
13: $\quad$ **if** $\frac{Inf^k}{b^k} \leq \beta$ **then**
14: $\quad\quad S(\beta) \leftarrow S(\beta) \setminus \{k\}$;
15: $\quad$ **end if**
16: **end while**
17: $A \leftarrow S(\beta)$;
//reward determination
18: $j^* \leftarrow arg\ max_{j \in S(\beta)} Inf^j; S(\beta)^* \leftarrow S(\beta) \setminus \{j^*\}$;
19: $q \leftarrow \frac{1}{2} \times \frac{Inf(S(\beta)) - \beta \times R}{min\{Inf^{j^*}, Inf(S(\beta)^*)\}}$;
20: **if** $Inf^{j^*} \leq Inf(S(\beta)^*)$ **then** $q^{j^*} \leftarrow \frac{1}{2} - q; q^T \leftarrow \frac{1}{2}$;
21: **else**
22: $\quad q^{j^*} \leftarrow \frac{1}{2}; q^T \leftarrow \frac{1}{2} - q$;
23: **end if**
24: **with probability** $q^{j^*} : r^{j^*} \leftarrow \frac{Inf^{j^*}}{\beta}$;
25: **with probability** $q^T : r^{j^*} \leftarrow R - \frac{Inf(S(\beta)^*)}{\beta}$;
26: **with probability** $q : r^{j^*} \sim Uniform[R - \frac{Inf(S(\beta)^*)}{\beta}, \frac{Inf^{j^*}}{\beta}]$;
27: **for** each $j \in S(\beta)^*$ **do**
28: $\quad r^j \leftarrow \frac{Inf^j}{Inf(S(\beta)^*)}(R - r^{j^*})$;
29: **end for**
30: **return** $(A, \mathbf{r})$;

In payment determination phase, we execute different reward rules to the agent $j^*$ with maximum influence and other agents, respectively. A random mechanism is applied to determine the reward of $j^*$ (Lines 24-26). We set the specific probabilities of $j^*$'s reward (Lines 19-23) to obtain the best approximation ratio. For other agents, the reward is allocated in proportion to the influence (Lines 27-29).

**Lemma 2**. *The agent selection of MTSS is computationally efficient.*

*Proof:* Computing the influence for all registered users (Lines 2-8) takes $O(n \times max_{j \in J} |SN^j \times m^2|)$ time, where computing the *Jaccard Similarity Coefficient* (Line 5) takes $O(m^2)$ time since there are at most $m$ tasks. Computing $\beta$ and $S(\beta)$ (Line 9) takes $O(n)$ time. The running time of while-loop (Lines 10-16) depends on the value of $\beta$. In each iteration of the while-loop, there are two possible value of $\beta$. If $\beta = \frac{Inf(S(\beta)) - max_{j \in S(\beta)} Inf^j}{R}$, this must be the last iteration of the while-loop. Otherwise, if $\beta = \frac{Inf^k}{b^k}$, the registered user $k$ will be removed from $S(\beta)$. Since there are at most $n$ registered users in $S(\beta)$, the while-loop thus takes $O(n)$ time. The running time of for-loop (Lines 27-29) takes $O(n)$ time. Thus, the running time of Algorithm 2 is bounded by $O(n \times max_{j\in J}|SN^j \times m^2|)$ [1]. □

Based on Lemma 2 and Theorem 5.2 in [57], we have the following Lemma.

**Lemma 3**. *The agent selection of MTSS is computationally efficient, individually rational, budget feasible, universally truthful, and has approximation ratio of 2.*

### 5.2 Online Budget Feasible Auction for Winner Selection

After agent selection, any social neighbor $i \in SN$ can bid for performing tasks. Since the social neighbors arrive in random order, we introduce the online budget feasible auction proposed by Zhao [10] to select the winners and determine the payment. The basic idea is dividing the time into multiple stages, and each stage has a fixed budget. Then the multiple-stage sampling accepting process is applied, where all departure social neighbors are samples. At the end of each stage, the density threshold is updated as the criterion of winner selection.

Illustrated in Algorithm 3, let $T = max\,\{e_1, e_2, ..., e_m\}$ be the deadline of whole crowdsourcing. The deadline is divided into $\lfloor log_2 T \rfloor + 1$ stages: $\{1, 2, ..., \lfloor log_2 T \rfloor, \lfloor log_2 T \rfloor + 1\}$. The stage $l$ ends at time step $T' = \lfloor 2^{l-1}T/2^{\lfloor log_2 T \rfloor} \rfloor$. Correspondingly, the stage budget for $l$-th stage is allocated as $B' = 2^{l-1}B/2^{\lfloor log_2 B \rfloor}$. We initialize the density threshold $\rho$ as a small constant $\epsilon$. At every time step $\boldsymbol{t}$, we add all new social neighbors arriving at step $\boldsymbol{t}$ to a set of online social neighbors $\mathscr{O}$ (Lines 3-5). Let $\mathscr{O}'$ be the set of unselected online social neighbors. We make a decision on whether to select unselected online social neighbors one by one in the order of their marginal values (Line 7). If the marginal density is not less than the current density threshold, and the allocated budget of this step has not been exhausted (Line 8), the social neighbor will be selected as a winner, and obtain the payment $V_i(S)/\rho$ (Line 9). If any social neighbor departs at time step $\boldsymbol{t}$ or any its bidding task expires at time step $\boldsymbol{t}$, we remove it from $\mathscr{O}$, and add it to the sample set $S'$ (Lines 15-17).

If time step $\boldsymbol{t}$ is an end time of a stage (Line 18), the density threshold will be updated by calling the function **DensityThreshold** according to the stage budget $B'$ and the sample set $S'$ (Line 19). Afterwards, we make a decision on whether to select online social neighbors base on the similar process shown in Lines 6-14, no matter whether they have ever been selected as the winners or not (lines 21-30). If the social neighbor can obtain a higher payment than before, according to the updated density threshold (Line 23), it will be selected as a winner with the new payment (Line 24).

Next, we give the **DensityThreshold** function, which is performed when the time step $\boldsymbol{t}$ is an end time of a stage. We adopt the *modified proportional share allocation rule* [52] according to stage budget $B'$ and the sample set $S'$. The key operation is selecting a winner set $\mathscr{G}$ based on the greedy

1. The running time of Algorithm 2 is very conservative since the number of agents is much less than $n$ in practice

**Algorithm 3 : Winner Selection of MTSS**

**Input:** task set $\Gamma$, bid profile $\mathbf{\Theta}$, budget $B$, deadline $T$
1: $(\boldsymbol{t}, T', \rho, S, S', B') \leftarrow (1, \frac{T}{2^{\lfloor log_2 T \rfloor}}, \epsilon, \emptyset, \emptyset, \frac{B}{2^{\lfloor log_2 B \rfloor}})$;
2: **while** $\boldsymbol{t} \le T$ **do**
3: **if** $a_i = \boldsymbol{t}$, for all $i \in SN$ **then**
4: $\mathscr{O} \leftarrow \mathscr{O} \cup \{i\}$; $\mathscr{O}' \leftarrow \mathscr{O} \setminus S$;
5: **end if**
6: **repeat**
7: $i \leftarrow arg\ max_{i' \in \mathscr{O}'} V_{i'}(S)$;
8: **if** $b_i \le \frac{V_i(S)}{\rho} \le B' - \sum_{i' \in S} p_{i'}$ **then**
9: $p_i = \frac{V_i(S)}{\rho}$; $S = S \cup \{i\}$;
10: **else**
11: $p_i \leftarrow 0$;
12: **end if**
13: $\mathscr{O}' \leftarrow \mathscr{O}' \setminus \{i\}$;
14: **until** $\mathscr{O}' = \emptyset$;
15: **if** $d_i = \boldsymbol{t}$ for all $i \in SN$ **then**
16: $\mathscr{O} \leftarrow \mathscr{O} \setminus \{i\}$; $S' \leftarrow S' \cup \{i\}$;
17: **end if**
18: **if** $\boldsymbol{t} = \lfloor T' \rfloor$ **then**
19: $\rho \leftarrow$ **DensityThreshold**$(B', S')$;
20: $T' \leftarrow 2T'$; $B' \leftarrow 2B'$; $\mathscr{O}' \leftarrow \mathscr{O}$;
21: **repeat**
22: $i \leftarrow arg\ max_{i' \in \mathscr{O}'} V_{i'}(S \setminus \{i'\})$;
23: **if** $b_i \le \frac{V_i(S\setminus\{i\})}{\rho} \le B' - \sum_{i' \in S} p_{i'} + p_i$ **and** $\frac{V_i(S\setminus\{i\})}{\rho} \ge p_i$ **then**
24: $p_i = \frac{V_i(S\setminus\{i\})}{\rho}$;
25: **if** $i \notin S$ **then**
26: $S = S \cup \{i\}$;
27: **end if**
28: **end if**
29: $\mathscr{O}' \leftarrow \mathscr{O}' \setminus \{i\}$;
30: **until** $\mathscr{O}' = \emptyset$;
31: **end if**
32: $\boldsymbol{t} \leftarrow \boldsymbol{t} + 1$;
33: **end while**

approach. As illustrated in Algorithm 4, the social neighbors in the sample set are sorted according to their marginal densities (Line 1). In this sorting, the $i$-th social neighbor is the social neighbor $i'$ such that $\frac{V'_i(\mathscr{G}_{i-1})}{b'_i}$ is maximum over all $i' \in S' \setminus \mathscr{G}_{i-1}$ (Line 4), where $\mathscr{G}_{i-1} = \{1, 2, \ldots, i-1\}$ and $\mathscr{G}_0 = \emptyset$. Considering the submodularity of value function $V$, this sorting implies that

$$\frac{V_1(\mathscr{G}_0)}{b_1} \ge \frac{V_2(\mathscr{G}_1)}{b_2} \ge ... \ge \frac{V_{|S'|}(\mathscr{G}_{|S'|-1})}{b_{|S'|}} \tag{16}$$

The set of winners is $\mathscr{G} = \{1, 2, \ldots, L\}$, where $L \le |S'|$ is the largest index such that its reserve price is no more than $\frac{V_i(\mathscr{G})B'}{V(\mathscr{G} \cup \{i\})}$. Finally, we set the density threshold as $V(\mathscr{G})/B'$.

Based on Theorem 2 in [10], we have the following lemma.

**Lemma 4**. *The winner selection of MTSS is computationally efficient, individually rational, budget feasible, truthful (cost-truthful and time-truthful), and constant competitiveness.*

The lemma 2, Lemma 3 and Lemma 4 together prove the following theorem.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3162108, IEEE Transactions on Mobile Computing



**Algorithm 4** :DensityThreshold

**Input:** stage budget $B'$, sample set $S'$

$\mathscr{G} \leftarrow \emptyset; i \leftarrow arg\ max_{i' \in S'} \frac{V'_i(\mathscr{G})}{b'_i}$;
**while** $b_i \leq \frac{V_i(\mathscr{G})B'}{V(\mathscr{G} \cup \{i\})}$ **do**
  $\mathscr{G} \leftarrow \mathscr{G} \cup \{i\}$;
  $i \leftarrow arg\ max_{i' \in S' \setminus \mathscr{G}} \frac{V'_i(\mathscr{G})}{b'_i}$;
**end while**
**return** $V(\mathscr{G})/B'$;

**Theorem 2**. *MTSS is computationally efficient, individually rational, budget feasible, cost-truthful, and time-truthful.*

## 6 Incentive Mechanism under Full-online Model

In this section, we present an *Incentive Mechanism for the Two-tiered Social Crowdsourcing System under Full-online Model (MTSF)*. *MTSF* also consists of two steps: agent selection and winner selection.

### 6.1 Agent Selection under Full-online Model

Differently from semi-online model, both of agents and social neighbors arrive in a random order. Note that the goal of agent selection is attracting more users from the social network to perform the tasks. To achieve the goal, we present the objective and constraint of agent selection:

- **Objective:** Since the agents arrive at platform dynamically, the cumulative online durations of the selected agents are desirable to cover the period of all the tasks [1, max$\{e_1, e_2, ..., e_m\}$] as much as possible.
- **Constraint:** The selected agents are expected to recruit the social neighbors who are interested in performing tasks.

The above objective of agent selection under full-online model is important. First, it will help to complete tasks as many as possible since the tasks are with different end time. Moreover, the dispersed online durations of agents prompt the social neighbors to bid for different unfinished tasks, especially the difficult tasks.

To optimize the objective, we define the coverage of any registered user $H^j, j \in J$, as the overlaps of the prediction of its online duration in [1, max$\{e_1, e_2, ..., e_m\}$]. We use Two-order Polynomial Fitting to predict the registered users' arrival time and departure time based on their history data.

Further, the marginal coverage of any registered user $j$ is defined as the overlap durations of $H^j$ and $H$, denoted as $H^j \cap H$, where $H$ is the uncovered time durations in the effective period of all tasks [1, max$\{e_1, e_2, ..., e_m\}$]. We tend to select the agents with large marginal coverage in order to achieve the objective.

To satisfy the constraint, we use the *Jaccard Similarity Coefficient Jac($\Gamma^j, i$)*, which has been given in subsection 5.1. Based on formula (13) and formula (14), we can calculate $Inf_i^j$ and $Inf^j$, respectively. Let $|H^j|$ be the number of time units in $H^j$. Then the unit influence of any registered user $j$ can be calculated as $Inf^j/|H^j|$. We tend to select the agents with large unit influence to satisfy the constraint.

Now, we propose our agent selection algorithm of *MTSF*, which follows a greedy approach. The basic idea of agent selection algorithm of *MTSF* is to select the registered users with maximum marginal coverage as the agents iteratively. Illustrated in Algorithm 5, we first calculate $Inf_i^j$ for $\forall j \in J, \forall i \in SN^j$ according to formula (13) (Lines 2-8). Then the registered users are sorted according to the marginal coverage (Line 11). In each iteration of agent selection, we select the registered user with the maximum marginal coverage over the uncovered time durations in [1, max$\{e_1, e_2, ..., e_m\}$], and check whether the marginal coverage is positive and the unit influence is larger than the threshold $\delta$ (Lines 10-16). If so, add it into the agent set $A$ (Line 13). $\delta$ is a predefined parameter determined by the platform. It reflects the desirable unit influence of the platform. The iteration terminates when the whole effective period of all tasks has been covered or $\gamma$ agents have been selected or all registered users have been processed (Line 10). Finally, we allocate a budget $\mathscr{B}^j$ to every selected agent $j$ in proportion to the influence over agent set $A$ (Line 18). Essentially, the budget is allocated based on the expected number of bidders influenced by agents. The similar allocation rule is applied to reward determination (Line 19).

**Algorithm 5 : Agent Selection of MTSF**

**Input:** registered users $J$, task set $\Gamma$, max number of agents $\gamma$, the budget $B$, total reward $R$

1: $A \leftarrow \emptyset$; $\gamma' \leftarrow \gamma$; $J' \leftarrow J$;
2: **for all** $j \in J'$ **do**
3:   $Inf^j \leftarrow 0$;
4:   **for all** $i \in SN^j$ **do**
5:     Calculate $Inf_i^j$ based on formula (13);
6:     $Inf^j \leftarrow Inf_i^j + Inf^j$;
7:   **end for**
8: **end for**
9: $H \leftarrow$[1, max$\{e_1, e_2, ...e_m\}$];
10: **while** $H \neq \emptyset$ **and** $\gamma' \neq 0$ **and** $J' \neq \emptyset$ **do**
11:   $j \leftarrow arg\ max_{h \in J' \setminus A}(H^h \cap H)$;
12:   **if** $H^j \cap H \neq \emptyset$ **and** $Inf^j/|H^j| > \delta$ **then**
13:     $A \leftarrow A \cup \{j\}$; $H \leftarrow H - H^j$; $\gamma' \leftarrow \gamma' - 1$;
14:   **end if**
15:   $J' \leftarrow J' \setminus \{j\}$;
16: **end while**
17: **for all** $j \in A$ **do**
18:   $\mathscr{B}^j = (Inf^j / \sum_{i \in A} Inf^i) \times B$;
19:   $r^j = (Inf^j / \sum_{i \in A} Inf^i) \times R$;
20: **end for**
21: **return** $(A, \mathscr{B}, \mathbf{r})$;

### 6.2 Winner Selection under Full-online Model

After agent selection, the agent will send a message, including the departure time, to the platform once it is online. The platform confirms the message. Specifically, if the agent's departure time is the same as some arrived agent's departure time, the platform will not select the agent as the winning agent. Then the winning agents will send the set of unexpired tasks to their social circles when they arrive at the platform. The influenced social neighbors then bid for the tasks through an online reverse auction.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3162108, IEEE Transactions on Mobile Computing



In this subsection, we design the winner selection algorithm of *MTSF* based on multiple-stage sampling accepting process. When any agent departs, a new stage begins, and the density threshold $\rho$ will be updated. Thus there are at most $\gamma$ stages.

Illustrated in Algorithm 6, we initialize the density threshold as a small constant $\epsilon$. For any step $\boldsymbol{t} \leq max\{e_1, e_2, ..., e_m\}$, if no agent departs at time step $\boldsymbol{t}$, the density threshold remains unchanged. Differently from the winner selection of *MTSS*, we should select winners for each agent. We process each agent $j \in A$. In each iteration, all new social neighbors of agent $j$ are added to a set of online social neighbor $\mathscr{O}^j$ (Lines 7-9). Then the similar online decision process in Algorithm 3 is used to select winners from unselected online social neighbor set $\mathscr{O}'^j$ for each agent $j$ (Lines 10-18). If the marginal density is not less than the current density threshold, and the allocated budget of agent $j$ has not been exhausted, the social neighbor will be selected as a winner (Lines 12-13). Finally, if any social neighbor departs at time step $\boldsymbol{t}$ or any of its bidding task expires at time step $\boldsymbol{t}$, we remove it from $\mathscr{O}^j$, and add it to the sample set $S'$. Meanwhile, the determination notices will be triggered (Lines 19-22).

If there is any agent $k$, who departs at time step $\boldsymbol{t}$, the density threshold will be updated (Line 26). Note that there is at most one such agent since any two agents have different departing times through the winning agent confirmation by the platform. The density threshold is computed by calling the function *DensityThreshold* (Algorithm 4) according to the lapsed budget $\mathscr{B}'$ and the sample set $S'$. The rest of the algorithm is same as Algorithm 3.

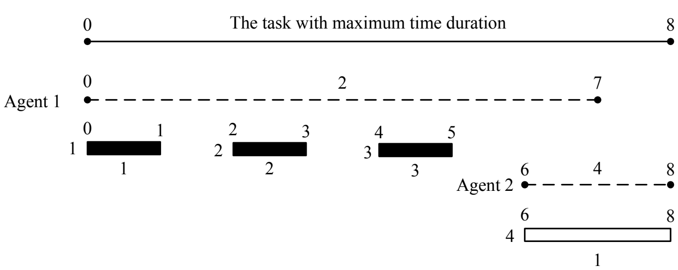



Fig. 5. An example illustrating how the winner selection algorithm of *MTSF* works, where the solid line represents the task with maximum time, the dotted lines represent the agent, the filled rectangles represent the social neighbors of agent 1, the hollow rectangle represents the social neighbor of agent 2. The numbers at both ends of the solid line represent the start time and end time of the task, respectively. The numbers at both ends of the dotted lines represent the arrival time and departure time of agents, respectively. The numbers above the dotted lines represent the allocated budgets of agents. The numbers beside the rectangles represent the IDs of social neighbors. The numbers at both ends of the rectangles represent the arrival time and departure time of social neighbors. The numbers below the rectangles represent the reserve price of social neighbors.

We use the example in Fig. 5 to show how the winner selection of *MTSF* works. In this example, the maximum time duration of all tasks is $[0, 8]$. $A = \{1, 2\}$, $(a^1, d^1, \mathscr{B}^1)$=(0, 7, 2), $(a^2, d^2, \mathscr{B}^2)$=(6, 8, 4) , $SN^1 = \{1, 2, 3\}$, $SN^2 = \{4\}$, $(a_1, d_1, b_1)$=(0, 1, 1), $(a_2, d_2, b_2)$=(2, 3, 2), $(a_3, d_3, b_3)$=(4, 5, 3), $(a_4, d_4, b_4)$=(6, 8, 1). $\Gamma_i, i \in \{1, 2, 3, 4\}$, can be omitted by assuming that each social neighbor has the same marginal value 1/2 when he is under consideration. For the sake of brevity, we consider that all tasks in $\Gamma_i$ will end after the

**Algorithm 6 : Winner Selection of MTSF**

**Input:** task set $\Gamma$, bid profile $\boldsymbol{\Theta}$, budget profile $\mathscr{B}$, agent set $A$

1: $(\boldsymbol{t}, \rho, S, S', \mathscr{B}') \leftarrow (1, \epsilon, \emptyset, \emptyset, 0)$;
2: **for all** $j \in A$ **do**
3: $\quad (S^j, \mathscr{O}^j, \mathscr{O}'^j) \leftarrow (\emptyset, \emptyset, \emptyset)$;
4: **end for**
5: **while** $\boldsymbol{t} \leq max\{e_1, e_2, ..., e_m\}$ **do**
6: $\quad$ **for all** $j \in A$ **do**
7: $\quad\quad$ **if** $a_i = \boldsymbol{t}$ for all $i \in SN^j$ **then**
8: $\quad\quad\quad \mathscr{O}^j \leftarrow \mathscr{O}^j \cup \{i\}$; $\mathscr{O}'^j \leftarrow \mathscr{O}^j \setminus S^j$;
9: $\quad\quad$ **end if**
10: $\quad\quad$ **repeat**
11: $\quad\quad\quad i \leftarrow arg\ max_{i' \in \mathscr{O}'^j} V_{i'}(S^j)$;
12: $\quad\quad\quad$ **if** $b_i \leq \frac{V_i(S^j)}{\rho} \leq \mathscr{B}^j - \sum_{i' \in S^j} p_{i'}$ **then**
13: $\quad\quad\quad\quad p_i = \frac{V_i(S^j)}{\rho}$; $S^j \leftarrow S^j \cup \{i\}$; $S \leftarrow S \cup \{i\}$;
14: $\quad\quad\quad$ **else**
15: $\quad\quad\quad\quad p_i \leftarrow 0$;
16: $\quad\quad\quad$ **end if**
17: $\quad\quad\quad \mathscr{O}'^j \leftarrow \mathscr{O}'^j \setminus \{i\}$;
18: $\quad\quad$ **until** $\mathscr{O}'^j = \emptyset$;
19: $\quad\quad$ **if** $d_i = \boldsymbol{t}$ **or** $e_k = \boldsymbol{t}$ for all $k \in \Gamma_i, i \in SN^j$ **then**
20: $\quad\quad\quad \mathscr{O}^j \leftarrow \mathscr{O}^j \setminus \{i\}$; $S' \leftarrow S' \cup \{i\}$;
21: $\quad\quad\quad$ Notify $i$ of the determination via agent $j$;
22: $\quad\quad$ **end if**
23: $\quad$ **end for**
24: $\quad$ **if** $d^k = \boldsymbol{t}$ for all $k \in A$ **then**
25: $\quad\quad \mathscr{B}' \leftarrow \mathscr{B}' + \mathscr{B}^k$;
26: $\quad\quad \rho \leftarrow$ **DensityThreshold**$(\mathscr{B}', S')$;
27: $\quad\quad$ **for all** $\mathscr{O}^j \neq \emptyset, j \in A$ **do**
28: $\quad\quad\quad \mathscr{O}'^j \leftarrow \mathscr{O}^j$;
29: $\quad\quad\quad$ **repeat**
30: $\quad\quad\quad\quad i \leftarrow arg\ max_{i' \in \mathscr{O}'^j} V_{i'}(S^j \setminus \{i'\})$;
31: $\quad\quad\quad\quad$ **if** $b_i \leq \frac{V_i(S^j \setminus \{i\})}{\rho} \leq \mathscr{B}^j - \sum_{i' \in S^j} p_{i'} + p_i$ **and** $\frac{V_i(S^j \setminus \{i\})}{\rho} \geq p_i$ **then**
32: $\quad\quad\quad\quad\quad p_i = \frac{V_i(S^j \setminus \{i\})}{\rho}$;
33: $\quad\quad\quad\quad\quad$ **if** $i \notin S^j$ **then**
34: $\quad\quad\quad\quad\quad\quad S^j = S^j \cup \{i\}$; $S = S \cup \{i\}$;
35: $\quad\quad\quad\quad\quad$ **end if**
36: $\quad\quad\quad\quad$ **end if**
37: $\quad\quad\quad\quad \mathscr{O}'^j \leftarrow \mathscr{O}'^j \setminus \{i\}$;
38: $\quad\quad\quad$ **until** $\mathscr{O}' = \emptyset$;
39: $\quad\quad$ **end for**
40: $\quad$ **end if**
41: $\quad \boldsymbol{t} \leftarrow \boldsymbol{t} + 1$;
42: **end while**

time step $d_i$. We set $\epsilon = 1/2$. Then the winner selection algorithm of *MTSF* works as follows.

- $\boldsymbol{t} = 0$: $S^1 = \emptyset$, $\rho = 1/2$, $b_1 = 1 \leq \frac{V_1(S^1)}{\rho} = 1 \leq \mathscr{B}^1 = 2$, thus $p_1 = 1$, $S = 1$.
- $\boldsymbol{t} = 2$: $S^1 = \{1\}$, $\rho = 1/2$, $b_2 = 2 > \frac{V_2(S^1)}{\rho} = 1$, thus $p_2 = 0$.
- $\boldsymbol{t} = 4$: $S^1 = \{1\}$, $\rho = 1/2$, $b_3 = 3 > \frac{V_3(S^1)}{\rho} = 1$, thus $p_3 = 0$.
- $\boldsymbol{t} = 6$: $S^2 = \emptyset$, $\rho = 1/2$, $b_4 = 1 \leq \frac{V_4(S^2)}{\rho} = 1 \leq \mathscr{B}^2 = 4$, thus $p_4 = 1$, $S = \{1, 4\}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3162108, IEEE Transactions on Mobile Computing



- $\boldsymbol{t} = 7$: $d^2 = t$, $S' = 1,2,3$, $\mathscr{B}^1 = 2$, update $\rho = 1/4$. $b_4 = 1 \leq \frac{V_4(S^2\backslash\{4\})}{\rho} = 2 \leq \mathscr{B}^2 - p_4 + p_4 = 4$, and $\frac{V_4(S^2\backslash\{4\})}{\rho} = 2 > p_4 = 1$, thus increase $p_4$ to 2.

Note that the payment to social neighbor 4 is increased from 1 to 2 by updating the density threshold when agent 1 departs.

### 6.3 Mechanism Analysis

In the following, we present the theoretical analysis, demonstrating that *MTSF* can achieve the desirable properties.

**Lemma 5**. *MTSF is computationally efficient.*

*Proof:* It suffices to prove that both Algorithm 5 and Algorithm 6 are computationally efficient.

In Algorithm 5, computing the influence for all registered users (Lines 2-8) takes $O(n \times max_{j\in J}|SN^j| \times m^2)$ time, where computing the *Jaccard Similarity Coefficient* (Line 5) takes $O(m^2)$ time since there are at most $m$ tasks. Finding the users with maximum marginal coverage takes $O(n)$ time. Since there are at most $n$ registered uses, the number of agents is at most $n$. Hence, the while-loop (Lines 10-16) takes $O(n^2)$ time. Therefore, the running time of agent selection is bounded by $O(max\{max_{j\in J}|SN^j|nm^2, n^2\})$[2].

In Algorithm 6, since the auction runs online, we only need to focus on the time complexity at each time step. The running time of the for-loop (Lines 6-23) is dominated by finding the social neighbor with maximum marginal value (Line 11). The time complexity of computing the marginal value is $O(max_{j\in A}|SN^j| \times |\Gamma_i|)$, where $|\Gamma_i|$ is at most $m$. Since there are $m$ tasks and each selected social neighbor should contribute at least one new task, the number of winners is at most $m$. Thus, the for-loop (Lines 6-23) takes $O(max_{j\in A}|SN^j|m^2)$ time. Next, we analyze the time complexity of the function **DensityThreshold** (Algorithm 4). Finding the social neighbor with the maximum marginal density takes $O(m|S'|)$ time, where $|S'|$ is at most $|SN|$. Since there are $m$ tasks and each selected social neighbor should contribute at least one new task, the number of winners is at most $m$. Thus, the running time of Algorithm 4 is $O(|SN|m^2)$. Finally, according to the similar analysis, the time complexity of selecting new winners from all online social neighbors (Lines 27-39) is $O(max_{j\in A}|SN^j|m^2)$. Hence, the running time of Algorithm 6 is bounded by $(O|SN|m^2)$. □

**Lemma 6**. *MTSF is individually rational.*

*Proof:* From the lines 12-16 and lines 31-32 of Algorithm 6, we can see that $p_i \geq b_i$ if any social neighbor $i$ is selected as a winner, otherwise $p_i = 0$. □

**Lemma 7**. *MTSF is budget feasible.*

*Proof: MTSF* allocates pro-rata budget $\mathscr{B}^j$ of total budget $B$ to each agent $j \in A$ according to the influence (Line 18 of Algorithm 5). From the lines 12-16 and lines 31-32 of Algorithm 6, we can see that it is guaranteed that the current total payment does not exceed its budget $\mathscr{B}^j$. Therefore, each agent is budget feasible, and when the agent $j$ departs, the total payment to the social neighbors of agent $j$ does not exceed $\mathscr{B}^j$. □

2. The running time of Algorithm 5 is very conservative since the number of agents is much less than $n$ in practice

**Lemma 8**. *MTSF is truthful (cost-truthful and time-truthful).*

*Proof:* Consider any social neighbor $i$ with a true bid is $\theta_i = (ra_i, rd_i, \Gamma_i, c_i, j)$ and the strategy bid $\hat{\theta}_i = (a_i, d_i, \Gamma_i, c_i, j)$. According to Algorithm 6, at each time step $\boldsymbol{t} \in [a_i, d_i]$, there may be a new decision on whether to accept social neighbor $i$, and at what price. We use $d^k_{\boldsymbol{t}}$, $B^j_{\boldsymbol{t}}$, $\rho_{\boldsymbol{t}}$, and $S^j_{\boldsymbol{t}}$ to represent the closest time step for updating density threshold (when agent $k$ departs), the residual budget of agent $j$, the current density threshold, and the selected social neighbors of agent $j$, respectively, at time step $\boldsymbol{t}$ before making decision on social neighbor $i$. Let $\hat{\theta}_{-i}$ be the strategy bid profile of all social neighbors excluding $i$.

We first prove that for fixed $b_i$ and $\hat{\theta}_{-i}$, reporting true arrival/departure time is a weakly dominant strategy for social neighbor $i$. According to Algorithm 6, social neighbor $i$ is paid for a price equal to the maximum price during $[a_i, d_i]$. Considering $ra_i \leq a_i \leq d_i \leq rd_i$, reporting $[a_i, d_i]$ would not help to obtain a higher payment for $i$.

Next, we prove that for fixed $[ra_i, rd_i]$, reporting the true cost is a weakly dominant strategy for social neighbor $i$. We first consider $i$ is selected as winner by reporting the true cost at time step $\boldsymbol{t} = ra_i$. In this case, there must be $c_i \leq V_i(S^j_{\boldsymbol{t}})/\rho_{\boldsymbol{t}} \leq \mathscr{B}^j_{\boldsymbol{t}}$, and $p_i = V_i(S^j_{\boldsymbol{t}})/\rho_{\boldsymbol{t}}$. If $i$ reports $b_i \leq V_i(S^j_{\boldsymbol{t}})/\rho_{\boldsymbol{t}}$, considering both $V_i(S^j_{\boldsymbol{t}})$ and $\mathscr{B}^j_{\boldsymbol{t}}$ are independent of $b_i$ in this case, $i$ wins still at time step $\boldsymbol{t} = ra_i$ with same payment. If $i$ reports $b_i > V_i(S^j_{\boldsymbol{t}})/\rho_{\boldsymbol{t}}$, he will lose at time step $\boldsymbol{t} = ra_i$, and $p_i = 0$.

Then, we consider the payment for $i$ in time duration $(\boldsymbol{t}, d^k_{\boldsymbol{t}})$ (determined by lines 10-18 of Algorithm 6). For any time step $\boldsymbol{t'} \in (\boldsymbol{t}, d^k_{\boldsymbol{t}})$, considering the submodularity of $V(S)$, there must be $V_i(S^j_{\boldsymbol{t'}}) \leq V_i(S^j_{\boldsymbol{t}})$. Note that the density threshold doesn't update in this case, i.e., $\rho_{\boldsymbol{t}} = \rho_{\boldsymbol{t'}}$. Therefore we have $p_i = V_i(S^j_{\boldsymbol{t'}})/\rho_{\boldsymbol{t'}} \leq V_i(S^j_{\boldsymbol{t}})/\rho_{\boldsymbol{t}}$ if $i$ is selected at $\boldsymbol{t'}$. Otherwise, $p_i = 0$. Therefore, a social neighbor cannot improve his payment by reporting false cost in time duration $[ra_i, d^k_{\boldsymbol{t}})$.

Next, we consider the payment for $i$ in time duration $[d^k_{\boldsymbol{t}}, rd_i]$ if $rd_i \geq d^k_{\boldsymbol{t}}$. For any time step $\boldsymbol{t'} \in [d^k_{\boldsymbol{t}}, rd_i]$, if $i$ reports $b_i \leq V_i\left(S^j_{\boldsymbol{t'}}\right)/\rho_{\boldsymbol{t'}} \leq \mathscr{B}^j_{\boldsymbol{t'}}$, he is still accepted with payment $V_i\left(S^j_{\boldsymbol{t'}}\right)/\rho_{\boldsymbol{t'}}$. If $i$ reports $b_i > V_i\left(S^j_{\boldsymbol{t'}}\right)/\rho_{\boldsymbol{t'}}$, he would not be selected at time step $\boldsymbol{t'}$. In this case, there may be other social neighbors to be selected at time step $\boldsymbol{t'}$, and the budget for agent $j$ will be diminished. Therefore, social neighbor $i$ cannot obtain higher payment in rest of time duration $(\boldsymbol{t'}, rd_i]$.

So far, we have proved that for fixed $[ra_i, rd_i]$, reporting the true cost is a weakly dominant strategy for social neighbor $i$ when he is selected as winner at time step $\boldsymbol{t} = ra_i$. Next, we consider the case when social neighbor $i$ is not a winner by reporting the true cost at time step $\boldsymbol{t} = ra_i$. In this case, there must be $c_i > V_i\left(S^j_{\boldsymbol{t}}\right)/\rho_{\boldsymbol{t}}$ or $V_i\left(S^j_{\boldsymbol{t}}\right)/\rho_{\boldsymbol{t}} > \mathscr{B}^j_{\boldsymbol{t}}$. In case $c_i > V_i\left(S^j_{\boldsymbol{t}}\right)/\rho_{\boldsymbol{t}}$, if social neighbor $i$ reports $b_i > V_i\left(S^j_{\boldsymbol{t}}\right)/\rho_{\boldsymbol{t}}$, then nothing changed. If social neighbor $i$ reports $b_i \leq V_i\left(S^j_{\boldsymbol{t}}\right)/\rho_{\boldsymbol{t}}$, he would win with payment $V_i\left(S^j_{\boldsymbol{t}}\right)/\rho_{\boldsymbol{t}}$ at time step $\boldsymbol{t}$. However, his utility will be negative. In addition, $\mathscr{B}^j_{\boldsymbol{t}'}$ remains unchanged in both above cases, and thus social neighbor $i$'s payment at time $\boldsymbol{t}' > \boldsymbol{t}$ is not affected. In case $V_i\left(S^j_{\boldsymbol{t}}\right)/\rho_{\boldsymbol{t}} > \mathscr{B}^j_{\boldsymbol{t}'}$, reporting a false cost does not affect the outcome at time step $\boldsymbol{t}$ or the residual budget $\mathscr{B}^j_{\boldsymbol{t}'}$ at time step $\boldsymbol{t}' > \boldsymbol{t}$. To sum up, reporting a false cost cannot improve social neighbor $i$'s payment. □

**Theorem 3**. *MTSF is computationally efficient, individually rational, budget feasible, cost-truthful, and time-truthful.*

## 7 Performance Evaluation

We have conducted thorough simulations to investigate the performance of *MTSO*, *MTSS*, and *MTSF*. We implemented three benchmark mechanisms:

- **Approximate optimal (offline):** the approximate optimal offline solution with full knowledge about agents and social neighbors. The problem is essentially a budgeted maximum coverage problem, which is a well-known NP-hard problem. It is known that a greedy algorithm provides $(\frac{e}{e-1})$ approximation solution [58]. Note that the approximate optimal mechanism is untruthful.
- **Intersection (full-online):** The coverage of registered users is defined as the overlaps of his online durations in the recent past. As illustrated in Fig. 6, Consider that the online durations in the past three days of any registered user $j$ are ([9:00, 13:00], [15:00, 19:00], [20:00, 23:00]), ([10:00, 14:00], [15:00, 18:00], [19:30, 23:00]), ([9:30, 12:30], [14:30, 17:30], [20:30, 22:30]), respectively. Then the coverage of $j$ is the common durations among the three days, i.e., $H^j$=([10:00, 12:30], [15:00, 17:30], [20:30, 22:30]).
- **Intermediaries (offline) [35]:** the offline mechanism that minimizes the intermediaries' total bid price with constraint of the number of data needed. For comparison with our mechanisms, we modify the constraint as the budget, i.e., selecting the users until the budget is consumed.
- **SocialRecruiter (offline) [56]:** SocialRecruiter maximizes the task completion with the limited budget by using the SIR epidemic model to model the task propagating and completing process.

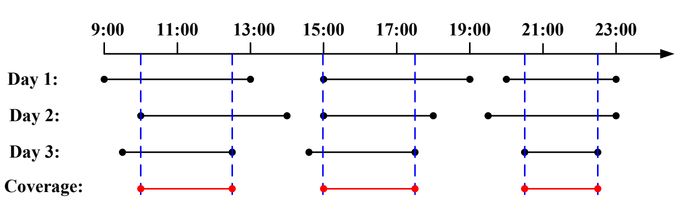



Fig. 6. Illustration for the coverage of any registered user $j$ through intersection.

We first measure the efficiency of agent selection phase. Then we measure the value with different number of agents, number of tasks, budgets and initial density threshold ($\epsilon$). Moreover, we measure the running time of *MSTF* and verify the truthfulness of *MSTF*. All the simulations were run on a Windows 10 machine with Intel(R) Core(TM) i5-8300H CPU and 8 GB memory. All algorithms are programed by Python 3.7. Each measurement is averaged over 100 instances.

### 7.1 Simulation Setup

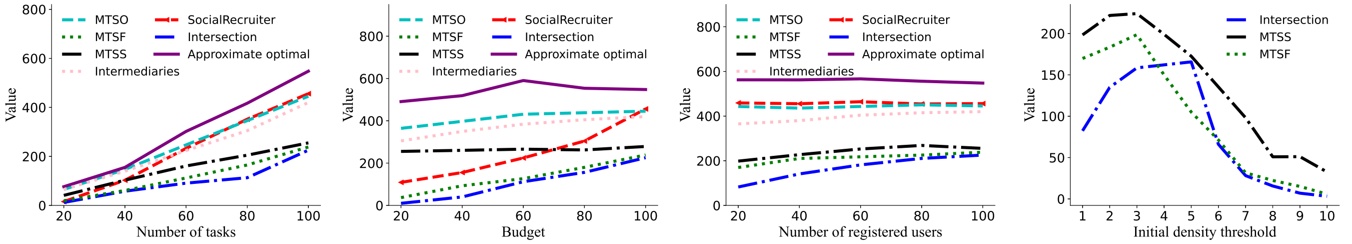
For our simulations, we use social circle data [59] from Facebook to simulate the relationship between agents and the users in social network. Facebook data was collected from survey participants using Facebook app. It includes node features (profiles), circles, and ego networks with 4039 nodes and 88234 edges. The arrival time and departure time of both registered users and social neighbors are derived from dataset [60], which consists of 410 workers. As the default setting, we choose 100 nodes from Facebook dataset as registered users and select 30 agents (only for full-online mechanisms) from the registered users with total reward of 100. Each task's end time is uniformly distributed over [1, 86400]. The cost of each bid of registered users and social neighbors is selected randomly from the auction dataset [61], which contains 5017 bid prices for Palm Pilot M515 PDA from eBay. The default number of tasks is 100. The budget of social neighbors is 100. The value of each task is uniformly distributed over [5, 8]. The initial density threshold ($\epsilon$) is 1. The maximum influence ($I_{max}$) is 1.2. The unit influence threshold ($\delta$) is $10^{-4}$. We will vary the value of key parameters to explore the impacts of these parameters.

### 7.2 Value

Fig. 7 compares the platform's value of all algorithms. From Fig. 7(a) and Fig. 7(b), we can see that the platform obtains a higher value when the number of tasks or the budget increases. As shown in Fig. 7(c), the platform's value also increases with the number of registered users since the platform can find better agents, thus more social neighbors can bid for performing crowdsourcing tasks. We can see from Fig. 7(d) that the platform's value of online mechanisms increases at beginning and then decreases when the initial density threshold increases. This is because when the initial density threshold increases, the platform may find social neighbors with lower bid and higher value. But if the initial density threshold is too high, the social neighbors are more difficult to be the winner and some of the tasks cannot be finished. This is because all winners should satisfy $V_i\left(S^j\right)/b_i \geq \rho$ according to Algorithm 3 and Algorithm 6. The approximate optimal, Intermediaries, SocialRecruiter and *MTSO* work in the offline manner, where the platform has the full knowledge about agents and users. Thus, these these offline mechanisms always outperform the *MTSS* (51.1% of approximate optimal on average) and *MTSF* (39.7% of approximate optimal on average). It is shown that *MTSO* sacrifices some performance (83.4% of approximate optimal on average) to achieve the cost-truthfulness compared with the approximate optimal mechanism. The performance of online mechanisms: *MTSO*, Intermediaries and SocialRecruiter are very close. In most cases, our *MTSF* obtains more value than that of intersection since the Two-order Polynomial Fitting provides a better prediction on the registered users' coverage than intersection (with average improvement of 19.2%). This is because Two-order Polynomial Fitting can reveal the tendency of historical data of

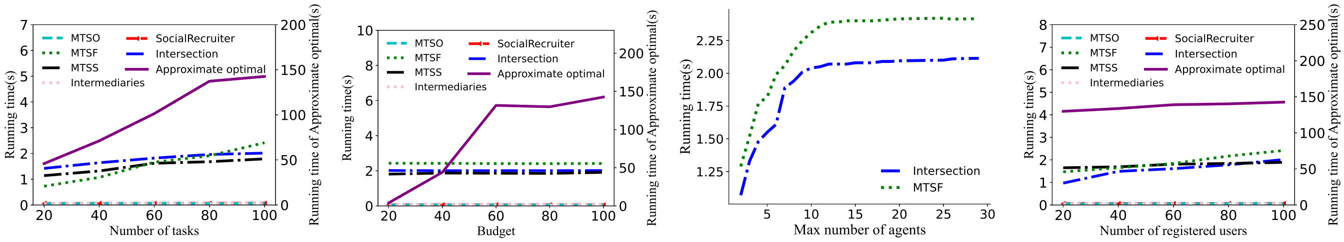
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3162108, IEEE Transactions on Mobile Computing





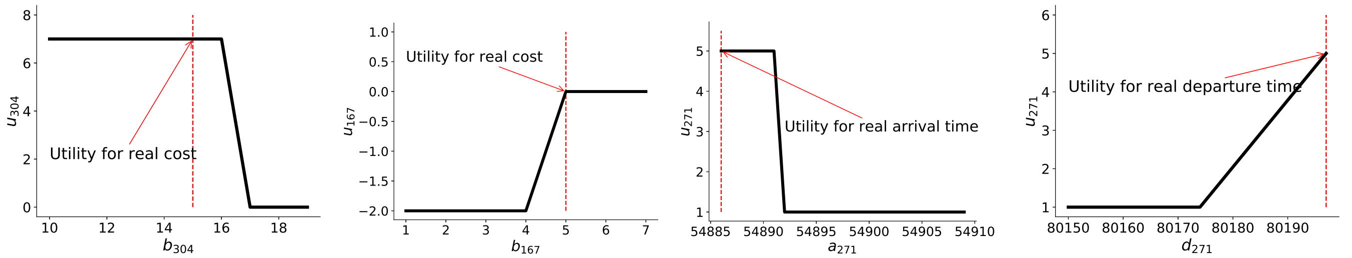

(a) (b) (c) (d)

Fig. 7. Value of the platform: (a) Value versus the number of tasks. (b) Value versus budget. (c) Value versus the number of registered users. (d) Value versus initial density threshold ($\epsilon$).




(a) (b) (c) (d)

Fig. 8. Running time: (a) Running time versus the number of tasks. (b) Running time versus budget. (c) Running time versus the max number of agents. (d) Running time versus the number of registered users




(a) (b) (c) (d)

Fig. 9. Cost-truthfulness and time-truthfulness of *MTSF*: (a) $c_{304} = 15$. (b) $c_{167} = 5$. (c) $ra_{271} = 54886$. (d) $rd_{271} = 80197$.

registered users'. Therefore, Two-order Polynomial Fitting can provide a prediction on the registered users' coverage. However, computing the intersection of historical coverage cannot provide the prediction. Besides, some extreme cases probably make the intersection method invalid, therefore decrease the performance of intersection method. For example, if one registered user arrives at 8:00 am and departs at 10:00 am in the first day, and he arrives at 10:00 am and departs at 12:00 am in the second day. If we use the intersection method, the coverage of this registered user will be empty. Moreover, as shown in Fig. 7(b), the value of SocialRecruiter increases sharply with the increase of budget. This is because the completing probability and propagating probability of users largely depend on the budget.

### 7.3 Running Time

Fig. 8 shows the running time of *MTSS* and *MTSF* with different number of tasks, budget, agents and registered users compared with other algorithms. It can be seen that the running time increase with the numbers of tasks, agents and registered users. This result is consistent with our analysis of computation complexity. In Fig. 8(b), the running time except the approximate optimal mechanism is stable when the budget increases. This is because the approximate optimal mechanism needs to compute the combinations of social neighbors under different budget, thus the running time increases while the budget increases. As shown in Fig. 8(c), when the number of agents increases, the number of social neighbors will increase accordingly. It leads to the increase of running time. The running time of *MTSF* and Intersection increase dramatically first and then tend stable when the max number of agents increases. This is because the selected agents have covered the period of all the tasks. From Fig. 8(a) and Fig. 8(d), we can see that when there are 100 registered users and 100 tasks, our incentive mechanisms can obtain the outcome within 2.4 seconds, which is much faster than the approximate optimal mechanism.

### 7.4 Truthfulness

We verified the cost-truthfulness of *MTSF* by randomly picking a winning social neighbor (ID=304) and a losing social neighbor (ID=167) and allowing them to bid prices that are different from their true costs. We illustrate the results in Fig. 9. We can see that social neighbor 304 achieves its optimal utility if it bids truthfully ($b_{304} = c_{304} = 15$) in Fig. 9(a) and social neighbor 167 achieves its optimal utility if it bids truthfully ($b_{167} = c_{167} = 5$) in Fig. 9(b). Then we further verified the time-truthfulness of *MSTF* by randomly picking one social neighbor (ID=271) and allowing him to report his arrival/departure times that are different from its true arrival/departure times. As shown in Fig. 9(c) and Fig. 9(d), social neighbor 271 achieves its

TABLE 2
Reward allocation rules for agent

| Goal | Criterion for offline model | Criterion for full-online model |
|---|---|---|
| registered user participation | | online time of registered user |
| social neighbor participation | number of selected social neighbors | influence of registered user [3] |
| task accomplish | value of selected social neighbors [3] | marginal coverage of registered user |
| comprehensive goal | marginal average value of selected social neighbors | average influence of registered user |

optimal utility if it reports its true arrival and departure times ($ra_{271} = a_{271} = 54886, rd_{271} = d_{271} = 80197$).

## 8 Discussion

This study proposed a two-tiered social crowdsourcing architecture to leverage the user influence in social network, and attract more participants. Essentially, the two-tiered social crowdsourcing architecture can extend the traditional crowdsourcing system by embedding the crowdsourcing in a social network through deep integration with social network technique.

From the view of system model, the character of two-tiered social crowdsourcing architecture is that there is an additional tier of agent between the crowdsourcing platform and participants. This poses the new challenges of incentive mechanism design for particular semi-online model and full-online model.

In order to enhance the applicability, we further discuss some closely related problems here.

- Decision of budget and reward

In practice, the budget and reward depends on many factors, such as resolve of crowdsourcer, market price, amount of funds, and crowdsouring cost. For example, the approach proposed in [62] decides the reward through the estimation of crowdsouring cost with different knowledge completeness level, and shows how the master can design an optimal contract by specifying different task-reward combinations for different user types. Such method can also be applied to our two-tiered social crowdsourcing architecture to decide the budget and reward. The study of budget or reward decision is out of the scope of this paper, and is an important topic in our future work. Note that the decision of budget and reward does not affect the desirable properties and performance of designed incentive mechanisms.

- Reward allocation rule

Since the registered users bid in semi-online model, we have designed the reward allocation of agent selection to avoid the strategic behavior by submitting the dishonest reserve. The reward allocation rule in semi-online model is determined by Algorithm 2 to achieve budget feasibility and universal truthfulness.

In this paper, we reward the agents based on the value of selected social neighbors to improve the task accomplish in offline model, and reward the agents based on the influence of registered users to improve the participation of social neighbors. However, since the registered users do not bid in both offline model and full-online model, we can apply other reward rules according to different goals we want to achieve, and the properties and performance of designed incentive mechanisms would not be affected. This enhances the flexibility of proposed incentive mechanisms. For example, in offline model, if the crowdsourcing platform expects to improve the participation of social neighbors, we can reward the agents based on the number of selected social neighbors. Alternatively, if the crowdsourcing platform expects to comprehensively improve the participation of social neighbors and task accomplish, we can reward the agents based on the marginal average value of selected social neighbors. We have given the possible goals and reward criterions in Table 2.

- Influence computation and diffusion

Some other types of influence computation methods can be considered in crowdsourcing systems, for example, history-based influence estimation [56], which explores the historical task diffusion events to estimate the influence to other social users.

Moreover, to reduce the knowledge required and time complexity for computing influence, we only take the social neighbors as the targets of task diffusion in this paper. However, if the crowdsourcing platform can obtain the globe knowledge of topology, interests, or diffusion history, we can recruit crowdsourcing participants not only from the social neighbors, but also from whole social network. Then, the global influence of agents to other social users can be calculated by employing some influence diffusion models. For example, Kempe *et al.* [63] proposed the two most popular influence diffusion models: independent cascade model and linear threshold model.

- Historical information

In this paper, we consider that the historical information is not always available, and only use the login information of registered users to estimate the online time. However, if the historical information is abundant and easy to obtain, both quantity and quality of users can be improved further. For examples, we can user the historical diffusion information to guide agent selection. Moreover, we can evaluate the quality of social users through the historical crowdsourcing results, and set the value of social users based on the quality. Thus, our incentive mechanisms are also applicable to quality-aware mechanisms.

## 9 Conclusion

In this paper, we have presented a two-tiered social crowdsourcing architecture to solve the insufficient participation

3. criterion used in this paper

problem in online mobile crowdsourcing systems by enabling the selected registered users to recruit more users from their social circles. We have presented three system models for our two-tiered social crowdsourcing system based on the availability of registered users and social neighbors: offline model, semi-online model, and full-online model, where the tasks are associated with different types and end times. We have presented an incentive mechanism for each of three system models with fixed budget and reward for task execution and task diffusion. Through both rigorous theoretical analysis and extensive simulations, we have demonstrated that the proposed incentive mechanisms achieve computational efficiency, individual rationality, budget feasibility and truthfulness, and the incentive mechanisms for semi-online model and full-online model can obtain averagely 51.1% and 39.7% value of approximate optimal untruthful offline algorithm, respectively.

## REFERENCES

[1] J. Xu, C. Guan, H. Wu, D. Yang, L. Xu, and T. Li, "Online incentive mechanism for mobile crowdsourcing based on two-tiered social crowdsourcing architecture," in *Proc. IEEE SECON*, 2018, pp. 1-9.
[2] Sensorly. (2021, Jun.) Sensorly. [Online]. Available: http://www.sensorly.com.
[3] P. Mohan, V. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," in *Proc. ACM SenSys*, 2008, pp. 323–336.
[4] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Toledo, J. Eriksson, S. Madden, and H. Balakrishnan, "Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones," in *Proc. ACM SenSys*, 2009, pp. 85–98.
[5] R. Rana, C. Chou, S. Kanhere, N. Bulusu, and W. Hu, "Earphone: An end-to-end participatory urban noise mapping," in *Proc. ACM/IEEE IPSN*, 2010, pp. 105–116.
[6] N. Maisonneuve, M. Stevens, M. Niessen, and L. Steels, "Noisetube: Measuring and mapping noise pollution with mobile phones," in *Proc. ITEE*, 2009, pp. 215–228.
[7] J. Xu, J. Xiang, and D. Yang. "Incentive mechanisms for time window dependent tasks in mobile crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6353-6364, 2015.
[8] J. Xu, Y. Zhou, Y. Ding, D. Yang, and L. Xu, "Bi-objective robust incentive mechanism design for mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 8, no. 19, pp. 14971-14984, 2021.
[9] L. Jiang, X. Niu, J. Xu, D. Yang, and L. Xu, "Incentive mechanism design for truth discovery in crowdsourcing with copiers," *IEEE Transactions on Services Computing*, https://doi.org/10.1109/TSC.2021.3075741, 2021.
[10] D. Zhao, X. Li, and H. Ma, "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint," in *Proc. IEEE INFOCOM*, 2014, pp.1213-1221.
[11] M. Xiao, J. Wu, L. Huang, R. Cheng, and Y. Wang, "Online task assignment for crowdsensing in predictable mobile social networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2306–2320, 2017.
[12] Y. Xiao, P. Simoens, P. Pillai, K. Ha, and M. Satyanarayanan, "Lowering the barriers to large-scale mobile crowdsensing," in *Proc. ACM HotMobile*, 2013, pp. 1-6.
[13] Yiguan Network Technology. (2017, Dec.) China's mobile map industry in the fourth quarter of 2016. [Online]. Available: https://www.sohu.com/a/126822301_162758
[14] Amazon Mechanical Turk. (2021, Jun.) Amazon Mechanical Turk. [Online]. Available: https://www.mturk.com/mturk/welcome
[15] Freelancer. (2021, Jun.) Freelancer. [Online]. Available: https://www.freelancer.com.
[16] Facebook. (2021, Jul.) Stepes. [Online]. Available: https://www.stepes.com/
[17] Google. (2021, Jul.) Image labeler. [Online]. Available: http://crowdsource.google.com/imagelabeler
[18] Google. (2021, Jul.) Translate Community. [Online]. Available: https://translate.google.com/community
[19] Tencent. (2021, Jul.) QQ-Crowd. [Online]. Available: http://task.qq.com/
[20] Baidu. (2021, Jul.) Crowdtesting. [Online]. Available: http://test.baidu.com/crowdtest/community/index
[21] Baidu. (2021, Jul.) Baidu Baike. [Online]. Available: https://baike.baidu.com/
[22] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proc. ACM MobiCom*, 2012, pp.173-184.
[23] Z. Mohammed, and M. Wagner (2014). "Data mining and analysis: Fundamental concepts and algorithms," Cambridge University Press. ISBN 9780521766333.
[24] I. Gaye, G. Mendy, S. Ouya, I. Diop, and D. Seck, "Multi-diffusion degree centrality measure to maximize the influence spread in the multilayer social networks," in *Proc. International Conference on e-Infrastructure and e-Services for Developing Countries*, 2016, pp. 53–56.
[25] J. Park, and B. Suh, "The impact of influential's betweenness centrality on the wom effect under the online social networking service environment," *Journal of Information Technology Applications and Management*, 2013, vol. 20, no. 2, pp. 127-146.
[26] M. Li, Q. Zhang, and Y. Deng, "Evidential identification of influential nodes in network of networks," *Chaos, Solitons Fractals*, vol. 117, pp. 283–296, 2018.
[27] R. I. Ogie. "Adopting incentive mechanisms for large-scale participation in mobile crowdsensing: from literature review to a conceptual framework," *Human-centric computing and information sciences*, vol. 6, no. 1, pp. 1-31, 2016.
[28] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao, "Incentives for mobile crowd sensing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 54-67, 2015.
[29] F. Restuccia, S. K. Das, and J. Payton, "Incentive mechanisms for participatory sensing: Survey and research challenges," *ACM Transactions on Sensor Networks*, vol. 12, no. 2, pp. 13-26, 2016.
[30] H. Gao, C. H. Liu, W. Wang, J. Zhao, Z. Song, X. Su, J. Crowcroft, and K. K. Leung, "A survey of incentive mechanisms for participatory sensing," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 918-943, 2015.
[31] S. Bhattacharjee, N. Ghosh, V. K. Shah, and S. K. Das, "QnQ: Quality and quantity based unified approach for secure and trustworthy mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 1, pp. 200-216, 2020.
[32] D. Yang, G. Xue, X. Fang, and J. Tang, "Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1732-1744, 2016.
[33] H. Jin, L. Su, D. Chen, H. Guo, K. Nahrstedt, and J. Xu, "Thanos: Incentive mechanism with quality awareness for mobile crowd sensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1951-1964, 2018.
[34] R. P. Barnwal, N. Ghosh, S. K. Ghosh, and S. K. Das, "PS-Sim: A framework for scalable data simulation and incentivization in participatory sensing-based smart city applications," *Pervasive and Mobile Computing*, vol. 57, pp. 64-77, 2019.
[35] Y. Chen, H. Chen, S. Yang, X. Gao, Y. Guo, and F. Wu, "Designing incentive mechanisms for mobile crowdsensing with intermediaries," *Wireless Communications and Mobile Computing*, 2019.
[36] H. Gao, C. Liu, J. Tang, D. Yang, P. Hui, and W. Wang, "Online quality-aware incentive mechanism for mobile crowd sensing with extra bonus," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2589-2603, 2018.
[37] J. Xu, Z. Rao, L. Xu, D. Yang, and T. Li, "Incentive mechanism for multiple cooperative tasks with compatible users in mobile crowd sensing via online communities," *IEEE Transactions on Mobile Computing*, vol. 19, no. 7, pp. 1618-1633, 2020.
[38] A. Di Stefano, M. Scatà, B. Attanasio, A. La Corte, P. Lió, and S. K. Das, "A novel methodology for designing Policies in Mobile Crowdsensing Systems," *Pervasive and Mobile Computing*, 67, 101230, 2020.
[39] J. Xu, G. Chen, Y. Zhou, Z. Rao, D. Yang, and C. Xie, "Incentive mechanisms for large-scale crowdsourcing task diffusion based on social influence," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3731-3745, 2021.
[40] Y. Chen, B. Li, and Q. Zhang, "Incentivizing crowdsourcing systems with network effects," in *Proc. IEEE INFOCOM*, 2016, pp.1-9.
[41] W. Wang, and Y. Jiang, "Multiagent-based allocation of complex tasks in social networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 4, pp. 571-584, 2015.
[42] P. Nguyen, and K. Nahrstedt, "Crowdsensing in opportunistic mobile social networks: A context-aware and human-centric approach," *arXiv*:1704.08598, 2017.

[43] Y. Han, T. Luo, D. Li, and H. Wu, "Competition-based participant recruitment for delay-sensitive crowdsourcing applications in D2D networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 12, pp. 2987-2999, 2016.

[44] T. Luo, S. S. Kanhere, and H. P. Tan, "SEW-ing a simple endorsement web to incentivize trustworthy participatory sensing" in *Proc. IEEE SECON*, 2014, pp.636–644.

[45] J. Nie, J. Luo, Z. Xiong, D. Niyato, and P. wang, "A stackelberg game approach toward socially-aware incentive mechanisms for mobile crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 724-738, 2018.

[46] J. Nie, Z. Xiong, D. Niyato, P. Wang, J. Luo, "A socially-aware incentive mechanism for mobile crowdsensing service market," in *Proc. IEEE GLOBECOM*, pp. 1-7 2018.

[47] J. Nie, J. Luo, Z. Xiong, D. Niyato, P. Wang, and H. V. Poor, "A multi-leader multi-follower game-based analysis for incentive mechanisms in socially-aware mobile crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1457-1471, 2020.

[48] G. Yang, S. He, Z. Shi, and J. Chen, "Promoting cooperation by the social incentive mechanism in mobile crowdsensing," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 86-92, 2017.

[49] J. Li, Z. Cai, M. Yan, and Y. Li, "Using crowdsourced data in location-based social networks to explore influence maximization," In *Proc. IEEE INFOCOM*, 2016, pp. 1-9.

[50] J. Sythoff, and J. Morrison. "Location-based services: Market forecast, 2011-2015," *Pyramid Research*, 2011.

[51] J. Schiller, and A. Voisard, "Location-based services," Elsevier, 2004.

[52] Y. Singer, "Budget feasible mechanisms," in *Proc. IEEE FOCS*, 2010, pp. 765–774.

[53] N. Chen, N. Gravin, and P. Lu, "On the approximability of budget feasible mechanisms," in *Proc. ACM SODA*, 2011, pp. 685-699.

[54] P. Jalaly, and E. Tardos, "Simple and efficient budget feasible mechanisms for monotone submodular valuations," *ACM Transactions on Economics and Computation*, vol. 9, no. 1, pp. 1-20, 2021.

[55] P. Jaccard, "The distribution of the flora in the alpine zone," *New phytologist*, vol. 11, no. 2, pp. 37-50, 1912.

[56] Z. Wang, Y. Huang, X. Wang, J. Ren, Q. Wang and L. Wu, "SocialRecruiter: Dynamic incentive mechanism for mobile crowdsourcing worker recruitment with social networks," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 2055-2066, 2021.

[57] N. Gravin, Y. Jin, P. Lu, and C. Zhang, "Optimal budget-feasible mechanisms for additive valuations," *ACM Transactions on Economics and Computation*, vol. 8, no. 4, pp. 1-15, 2020.

[58] S. Khullera, A. Mossb, and J. Naor, "The budgeted maximum coverage problem," *Information Processing Letters*, vol. 70, pp. 39–45, 1999.

[59] J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks," in *Proc. NIPS*, 2012, pp. 539-547.

[60] OSCHINA. (2021, Jun.) Time Dataset. [Online]. Available: https://gitee.com/luozhuangye/mtsf/blob/master/data.csv

[61] W. Jank, and G. Shmueli. (2021, Jun.) Modeling Online Auctions. [Online]. Available: http://www.modelingonlineauctions.com/datasets

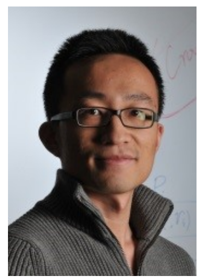
[62] L. Duan, T. Kubo, K. Sugiyama, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand, "Motivating smartphone collaboration in data acquisition and distributed computing," *IEEE Transactions on Mobile Computing*, vol. 13, no. 10, pp. 2320-2333, 2014.

[63] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proc. SIGKDD*, 2003, pp. 137-146.


**Jia Xu** (*SM*'21) received the M.S. degree in School of Information and Engineering from Yangzhou University, Jiangsu, China, in 2006 and the PhD. degree in School of Computer Science and Engineering from Nanjing University of Science and Technology, Jiangsu, China, in 2010. He is currently a professor in the School of Computer Science at Nanjing University of Posts and Telecommunications. His main research interests include crowdsourcing, edge computing and wireless sensor networks.

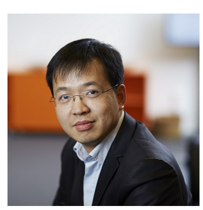
**Zhuangye Luo** received the bachelor's degree in School of Computer Science from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2019. He is currently working toward the master's degree in the same university. His research interests are mainly in the areas of the mobile crowd sensing and machine learning.

**Chengcheng Guan** received the bachelor's degree in School of Computer Science from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2016. He is currently working toward the master's degree in the same university. His research interests are mainly in the areas of the mobile crowd sensing, wireless sensor networks, and machine learning.



**Dejun Yang** (*SM*'19) received the B.S. degree in computer science from Peking University, Beijing, China, in 2007 and the Ph.D. degree in computer science from Arizona State University, Tempe, AZ, USA, in 2013. Currently, he is an Associate Professor of computer science with Colorado School of Mines, Golden, CO, USA. His research interests include Internet of things, networking, and mobile sensing and computing with a focus on the application of game theory, optimization, algorithm design, and machine learning to resource allocation, security and privacy problems.

**Linfeng Liu** (*M*'13) received the B. S. and Ph. D. degrees in computer science from the Southeast University, Nanjing, China, in 2003 and 2008, respectively. At present, he is a Professor in the School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, China. His main research interests include the areas of vehicular ad hoc networks, wireless sensor networks and multi-hop mobile wireless networks.



**Yan Zhang** (*F*'19) received the Ph.D. degree from the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. He is currently a Full Professor with the Department of Informatics, University of Oslo, Oslo, Norway. His research interests include next-generation wireless networks leading to 5G beyond/6G, green and secure cyber-physical systems (e.g., smart grid and transport). Dr. Zhang is an IEEE Vehicular Technology Society Distinguished Lecturer. He is a Fellow of IET. He is the Chair of IEEE Communications Society Technical Committee on Green Communications and Computing. Prof. Zhang was a recipient of the Highly Cited Researcher Award (Web of Science top 1% most cited) by Clarivate Analytics.