Online Incentive Mechanism for Mobile Crowdsourcing based on Two-tiered Social Crowdsourcing Architecture

Jia Xu[1], Chengcheng Guan[1], Haobo Wu[1], Dejun Yang[1,2], Lijie Xu[1], Tao Li[1]
[1]Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023 China
[2]Department of Computer Science, Colorado School of Mines, Golden, CO 80401 USA
Email: xujia@njupt.edu.cn, 1216043121@njupt.edu.cn, njuptwhb@163.com, djyang@mines.edu, ljxu@njupt.edu.cn, towerlee@njupt.edu.cn

***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 in 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. In the two-tiered social crowdsourcing system, the tasks are associated with different end times, and both the registered users and their social neighbors have different arrival/departure times. An online incentive mechanism, *MTSC,* which consists of two steps: *Agent Selection* and *Online Reverse Auction*, is proposed for this novel mobile crowdsourcing system. Through both rigorous theoretical analysis and extensive simulations, we demonstrate that the proposed incentive mechanism achieves computational efficiency, individual rationality, budget feasibility, cost truthfulness, and time truthfulness.**

Keywords—mobile crowdsourcing; incentive mechanism; social network; online

I. 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 [1] for constructing cellular/WiFi network coverage maps, Nericell [2] and VTrak [3] for providing traffic information, as well as Ear-Phone [4] and NoiseTube [5] for creating noise maps.

The incentive mechanisms are crucial for mobile crowdsourcing systems to compensate participants' resource consumption and potential privacy threats. The incentive mechanisms also help to improve service quality since the services are truly dependent on the quantity of users and the quality of crowdsourcing data.

There have been many research efforts on designing incentive mechanisms for mobile crowdsourcing systems [6, 7]. Online incentive mechanisms [8, 9] aim to deal with the mobile crowdsourcing, where the users arrive one by one in random order and user availability will change over time. The online incentive mechanisms enable the decision on whether to buy the 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. There are two examples:

- Insufficient active users

According to the data of the fourth quarter in 2016 from Analysys [10], 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.

- Insufficient qualified users

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) [11] requires the workers to complete a test in order to be qualified. We observed that there were 618.65 uncompleted HITs per day on average from 2016-05-01 to 2016-05-20. Moreover, 91.75 HITs would be expired within one hour per day on average.

To address the insufficient participation problem, we extend the mobile crowdsourcing systems to the social networks in order to recruit more participants. Nowadays, the social network is evolving and integrating with many aspects of our lives. The celebrities usually show significant influence in the social network. In Kickstarter crowdsourcing platform [12], the American musician Amanda Palmer launched a crowdsourcing project for the new CD and concert plan with target 0.1 million dollars. She got more than 1.19 million dollars ultimately. The American actress Kristen Bell launched a crowdsourcing project for the new film "Veronica Mars". Ultimately, she got more than 5.7 million dollars from the fans, comparing with the target of 2 million dollars.

In this paper, we propose a two-tiered social crowdsourcing architecture. In the proposed crowdsourcing system, a set of agents are selected from the registered users of the crowdsourcing system. The agents arrive at platform dynamically. Once they are online, the agents will diffuse the crowdsourcing tasks to their social neighbors through the social circle, such as Twitter, Microblog, Facebook, and WeChat. We model the task allocation process as a sealed

978-1-5386-4281-8/18/$31.00 ©2018 IEEE

online reverse auction. The platform publicizes tasks with a budget. The social neighbors who are interested in performing the tasks can bid via the agents for participating crowdsourcing tasks. We consider that the social neighbors also arrive at the platform dynamically. Thus, the platform should decide whether to buy the users' service before the departure of both the social neighbors and the agents. The winners then perform the tasks and submit the results to the platform via the agents. Finally, each winner obtains the payment, which is determined by the platform. The objective of our incentive mechanism is designing truthful incentive mechanisms to maximize the value from the winners' services under the budget constraint [14]. The whole process is illustrated by Fig.1. Different from most existing mobile crowdsourcing systems, there are two interaction tiers in the crowdsourcing system: the interaction between the platform and the agents, and the interaction between the agents and their social neighbors.

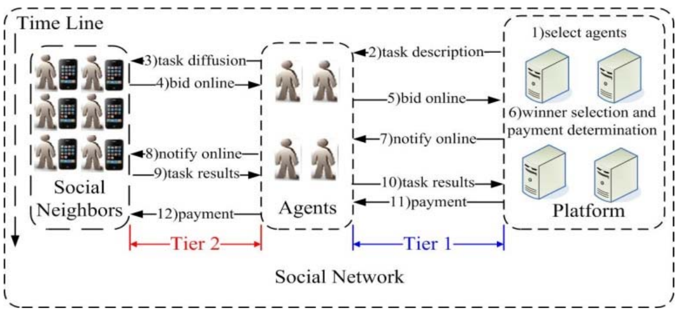


Fig. 1 System model of the two-tiered social crowdsourcing system

The problem of designing truthful incentive mechanism for the two-tiered social crowdsourcing system is very challenging. First, a realistic system model should be defined to formalize 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. Moreover, the agents should be selected to satisfy certain desirable goals, such as maximizing the online durations or maximizing the influence to their social neighbors. Finally, the social neighbors may take a strategic behavior by submitting dishonest bid price and arrival/departure time to maximize their utility.

The main contributions of this paper are as follows:

- We present a two-tiered social crowdsourcing architecture to solve the insufficient participation problem using the social network in online scenario.
- We propose the *Agent Selection* algorithm based on the historical information and personal profiles of the social network to optimize the online duration coverage and the unit influence simultaneously.
- We design the *Online Reverse Auction* for selecting the social neighbors and calculating payments. We show that the designed auction satisfies the desirable properties of computational efficiency, individual rationality, budget feasibility, and truthfulness.

The rest of the paper is organized as follows. Section II formulates two system models and lists some desirable properties. Section III presents the detailed design of our incentive mechanism. Section IV presents the analysis of our incentive mechanism. Performance evaluation is shown in section V. We review the state-of-art research in Section VI, and conclude this paper in Section VII.

II. System Model And Desirable Properties

A. *System Model of Two-tiered Social Crowdsourcing*

We consider that the crowdsourcing platform is owned by an online community. Thus the platform can extract the personal profile of users in the online community. This assumption is reasonable since many online communities have developed crowdsourcing systems themselves, such as Stepes [15] owned by Facebook, Google Image Labeler [16] and Translate Community [17] owned by Google+, QQ-Crowd [18] owned by QQ, Crowdtesting [19] and Baidu Baike [20] owned by Baidu.

Assume that a set of registered users $J = \{1,2, \dots, n\}$ of the platform are interested in diffusing crowdsourcing tasks. The platform first selects γ agents from all registered users. Each agent chosen by the platform can get a reward from the platform. We denote the agent set as $A = \{1,2, \dots, \gamma\}$, where γ depends on the total reward λ of the platform.

Then the platform publicizes a set of tasks $\Gamma = \{\tau_1, \tau_2, \dots, \tau_m\}$, where task $\tau_k, k = 1, 2, \dots m$, is associated with an end time e_k and a type t_k. For the sake of brevity, we consider all tasks are launched at time zero. 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. 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.

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, $\max\{e_1, e_2, \dots, e_m\} \geq d^j \geq a^j \geq 0$. The reward to any agent is proportional to its online duration. Thus the agents will state their real arrival/departure time since it is impossible to state an early arrival time or late departure time in practice.

Once any agent j is online, he/she will send a message $MSG = (\Gamma^j, a^j,\ d^j)$, where Γ^j is the set of unexpired tasks when j arrives at the platform, to the platform. After confirmed by the platform, the agent j sends the same message to its social circle, such as Twitter, Facebook, and WeChat, in order to diffuse the tasks. We use the set SN^j to represent the set of influenced social neighbors of agent j. We use SN to represent the set of influenced social neighbors of all agents, i.e., $SN = \bigcup_{j \in A} SN^j$.

After task diffusion, the influenced social neighbors can participate in the mobile crowdsourcing through an online reverse auction, and each social neighbor can get the payment for providing the service. In the online reverse auction, 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. In bid θ_i, a_i and d_i are the arrival time and departure time of social neighbor i, respectively. 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 j. $\Gamma_i \subseteq \Gamma$ is the task set he/she is willing to perform, and b_i is the reserve price.

When any bid θ_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.

We consider the real cost c_i for performing Γ_i, the real arrival time ra_i, and departure time rd_i are private and unknown to other social neighbors, agents and the platform. Since we consider the social neighbors are selfish and rational individuals, each social neighbor can behave strategically by submitting the dishonest reserve price, arrival/departure time to maximize its utility. Note that we assume that a social neighbor cannot announce an earlier arrival time or a later departure time than his/her true arrival/departure time, i.e., $ra_i \leq a_i \leq d_i \leq rd_i$. This assumption is justified since the presence can be directly verified by the platform. For the same reason, a social neighbor cannot lie about the task set he/she is willing to perform and the agent he is associated with.

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

$$u_i = p_i - c_i . \quad (1)$$

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.

We consider an incentive mechanism $\mathcal{M}(\alpha, \mathcal{w}, p)$ consisting of an agent selection function α, a winner selection function $\mathcal{w}$, and a payment decision function p. For any registered user set J, the task set Γ, and the desirable number of agents γ, the function $\alpha(J, \Gamma, \gamma)$ outputs the subset of registered users $A \subseteq J$. For any budget B, the task set Γ and the bid profile $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_{|SN|})$, the function $\mathcal{w}(B, \Gamma, \boldsymbol{\theta})$ outputs the subset of influenced social neighbors $S \subseteq SN$, the function $p(B, \Gamma, \boldsymbol{\theta})$ returns a vector $\mathbf{p} = (p_1, p_2, \ldots, p_{|SN|})$ of payments to all the influenced social neighbors. 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, i.e.,

$$\text{Maximize } V(S) \text{ subject to } \sum_{i \in S} p_i \leq B$$

In this study, we focus on the case where $V(S)$ is nonnegative, monotone, and submodular. Submodularity has been applied in many papers [4, 22]. This covers many realistic scenarios, such as [2, 21].

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

$$f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y)$$

for any $X \subseteq Y \subseteq \Omega$ *and* $x \in \Omega \setminus Y$, *and it is monotone if and only if* $f(X) \leq f(Y)$, *where* 2^{Ω} *is the power set of* Ω, $\mathbb{R}$ *is the set of reals.*

B. Desirable Properties

Our objective is to design an incentive mechanism satisfying the following desirable properties:

- **Computational efficiency:** An incentive mechanism is computationally efficient if the agent set A, the winner set S, and the payment can be computed in polynomial time.
- **Individual Rationality:** Each social neighbor will have a non-negative utility while reporting the true cost, and the arrival/departure time, i.e., $u_i \geq 0, \forall i \in SN$.
- **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, arrival/departure time, no matter what others submit. In other words, reporting the real cost and arrival/departure time is a weakly dominant strategy for all users.

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

III. Incentive Mechanism Design

In this section, we present an *Online Incentive Mechanism for the Two-tiered Social Crowdsourcing System* (*MTSC*). *MTSC* consists of two steps: *Agent Selection* for selecting agents from the registered users and *Online Reverse Auction* for selecting winners and calculating payments.

A. Agent Selection

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 for *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 $[0, \max\{e_1, e_2, \ldots, e_m\}]$ as much as possible.
- **Constraint:** The selected agents are expected to recruit the social neighbors who are interested in performing tasks.

To optimize the objective, we define the coverage of any registered user $\mathcal{H}^j$, $j \in J$, as the overlaps of his online durations in the recent past in $[0, \max\{e_1, e_2, \ldots, e_m\}]$. As illustrated in Fig.2, 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. Since we consider that the crowdsourcing platform rides on an online community, the historical log-in/out time of each registered user can be extracted from the personal profile of the online community. Then the coverage of j is the common durations among the three days, i.e., $\mathcal{H}^j$=([10:00, 12:30], [15:00, 17:30], [20:30, 22:30]).

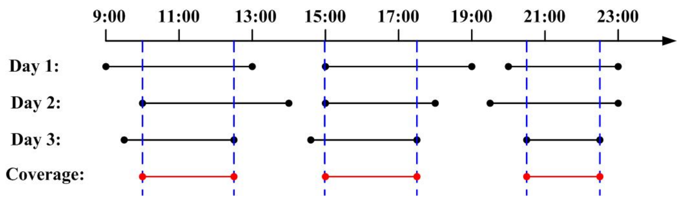


Fig. 2 Illustration for the coverage of any registered user j.

Further, the marginal coverage of any registered user j is defined as $\mathcal{H}^j \cap \mathcal{H}$, where $\mathcal{H}$ is the uncovered time durations in the effective period of all tasks $[0, \max\{e_1, e_2, \dots, e_m\}]$. 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* $Jac(\Gamma^j, i)$ [23] to measure how well the types of Γ^j match the interests of any social neighbor i:

$$Jac(\Gamma^j, i) = \frac{|T^j \cap I_i|}{|T^j \cup I_i|},$$

where T^j is the set of types of tasks in Γ^j, I_i is the interests of social neighbor i, which can also be extracted from the personal profile of the online community.

Considering the social neighbor's diminishing return on the influence of registered user, we introduce the influence function originated from task influence maximization [22]:

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

where Z is the input influence parameter 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 $\frac{Z^2 I(Z, I_{max})}{\partial Z^2} < 0$ for $Z \in (0,1)$.

Then the influence of any registered user j to any social neighbor i is defined as

$$Inf_i^j = I(Jac(\Gamma^j, i), I_{max}) \,. \tag{2}$$

We use Inf_i^j to measure the possibility of i bidding for performing any task in Γ^j when any registered user j diffuses the task set Γ^j to any social neighbor i.

The influence of any registered user j can be calculated as $Inf^j = \sum_{i \in SN^j} Inf_i^j$. The unit influence of any registered user j can be calculated as $Inf^j / |\mathcal{H}^j|$. We tend to select the agents with large unit influence to satisfy the constraint.

Thus far, we have proposed two new metrics: marginal coverage and unit influence for optimizing the objective and satisfying the constraint, respectively. Now, we propose our algorithm of *Agent Selection*, which follows a greedy approach. Illustrated in Algorithm 1, we first calculate Inf_i^j for $\forall j \in J, \forall i \in SN^j$ according to formula (2). Then the registered users are sorted according to the marginal coverage. In each iteration of *Agent Selection*, we select the registered user with the maximum marginal coverage over the uncovered time durations in $[0, \max\{e_1, e_2, \dots, e_m\}]$, and check whether the marginal coverage is positive and the unit influence is larger than the constant δ. If so, add him/her into the agent set A. δ 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 γ agents have been selected or all registered users have been processed. Finally, we allocate a budget $\mathcal{B}^j$ to any selected agent $j \in A$ in proportion to the influence over agent set A.

Algorithm 1: Agent Selection

Input: registered users J, task set Γ, desirable number of agents γ, the budget B

```
A ← ∅; γ' ← γ; J' ← J;
for all j∈J' do
    Inf^j ← 0;
    for all i ∈ SN^j do
        Calculate Inf_i^j based on formula (2);
        Inf^j ← Inf^j + Inf_i^j;
    end for
end for
H ← [0, max{e_1, e_2, ..., e_m}];
while H ≠ ∅ and γ' ≠ 0 and J' ≠ ∅ do
    j ← arg max_{h∈J'\A} (H^h ∩ H);
    if (H^j ∩ H) > 0 and Inf^j/|H^j| > δ then
        A ← A ∪ {j}; J' = J'\{j}; H ← H − H_j; γ' ← γ' − 1;
    end if
end while
for all j ∈ A do
    B^j = (Inf^j / Σ_{i∈A} Inf^i) × B;
end for
return (A, B);
```

B. Online Reverse Auction

After *Agent Selection*, the agent will send a message, including the departure time, to the platform once he/she 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.

In this subsection, we design the algorithm of *Online Reverse Auction* based on *multiple-stage sampling accepting* process. Different from the existing work [8], the *stage-budgets* are allocated to the agents. When any agent departs, a new stage begins, and the density threshold ρ will be updated. Thus there are at most γ stages. Illustrated in Algorithm 2, the *Online Reverse Auction* outputs the set of winning social neighbors S and the payment vector $\mathbf{p}$ of all influenced social neighbors.

We initialize the density threshold as a small constant ϵ. For any step $t \le \max\{e_1, e_2, \dots, e_m\}$, if no agent departs at time step t, the density threshold remains unchanged. We process each agent $j \in A$. In each iteration, all new social neighbors of agent $j \in A$ are added to a set of online social neighbor O^j(Lines 7-9). Then we make a decision on whether to select unselected online social neighbors one by one in the order of their marginal values (Lines 10-16). Given a set of selected social neighbors of agent S^j, the marginal value of social neighbor $i \notin S^j$ is $V_i(S^j) = V(S^j \cup \{i\}) - V(S^j)$, and the marginal density is $V_i(S^j)/b_i$. The social neighbors with higher marginal values will be considered as winner first. 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, and obtain the payment $V_i(S^j)/\rho$ (Line 13). Finally, if any social neighbor departs at time step t or any his/her bidding task expires at time step t, we remove he/she from $\mathcal{O}^j$, and add he/she to the sample set S'. Meanwhile, the determination notices will be triggered (Lines 17-20).

Algorithm 2: Online Reverse Auction

Input: task set Γ, bid profile $\boldsymbol{\theta}$, budget profile $\boldsymbol{\mathcal{B}}$, agent set A

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

If there is any agent k, who departs at time step t, the density threshold will be updated (Line 23). Note that there is at most one such agent since any two agents in the winning agent set have different departing times through the winning agent confirmation by the platform. The density threshold is computed by calling the *DensityThreshold* function (to be elaborated later) according to the allocated budget of agent k and the sample set S'. Afterwards, we process each agent $j \in A$, who still has online social neighbors (Lines 24-34). In each iteration, we make a decision on whether to select these online social neighbors base on the similar process shown in Lines 10-16, no matter whether they have ever been selected as the winners or not. If the social neighbor can obtain a higher payment than before, according to the updated density threshold, he/she will be selected as a winner with the new payment (Line 29).

Algorithm 3: DensityThreshold

Input: agent k's budget $\mathcal{B}^k$, sample set S'

$\mathcal{G} \leftarrow \emptyset$; $i \leftarrow \arg\max_{j \in S'} \frac{V_j(\mathcal{G})}{b_j}$;
while $b_i \le \frac{V_i(\mathcal{G})\mathcal{B}^k}{V(\mathcal{G} \cup \{i\})}$ **do**
$\mathcal{G} \leftarrow \mathcal{G} \cup \{i\}$;
$i \leftarrow \arg\max_{j \in S' \backslash \mathcal{G}} \frac{V_j(\mathcal{G})}{b_j}$;
end
return $V(\mathcal{G})/\mathcal{B}^k$;

Next, we give the *DensityThreshold* function, which is performed when any agent k departs at time step t. We adopt the *proportional share allocation rule* [24] according to agent k's budget $\mathcal{B}^k$ and the sample set S'. The key operation is selecting a winner set $\mathcal{G}$, which follows a greedy approach. As illustrated in Algorithm 3, the social neighbors in the sample set are sorted according to their marginal densities. In this sorting, the ith social neighbor is the social neighbor j such that $\frac{V_j(\mathcal{G}_{i-1})}{b_j}$ is maximum over $S' \backslash \mathcal{G}_{i-1}$, where $\mathcal{G}_{i-1} = \{1, 2, \dots, i-1\}$ and $\mathcal{G}_0 = \emptyset$. Considering the submodularity of value function V, this sorting implies that

$$\frac{V_1(\mathcal{G}_0)}{b_1} \ge \frac{V_2(\mathcal{G}_1)}{b_2} \ge \cdots \ge \frac{V_{|S'|}(\mathcal{G}_{|S'|-1})}{b_{|S'|}}$$

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

We use the example in Fig.3 to show how the *Online Reverse Auction* works. In this example, the maximum time duration of all tasks is [0, 8]. $A = \{1, 2\}$, $(a^1, d^1, \mathcal{B}^1) = (0, 7, 2)$, $(a^2, d^2, \mathcal{B}^2) = (6, 8, 4)$, $SN^1 = \{1, 2, 3\}$, $SN^2 = \{4\}$, $(a_1, d_1, b_1) = (0, 1, 1)$, $(a_2, d_2, b_2) = (2, 3, 2)$, $(a_3, d_3, b_3) = (4, 5, 3)$, $(a_4, d_4, b_4) = (6, 8, 1)$. $\Gamma_i, i \in \{1, 2, 3, 4\}$ 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 Γ_i will end after the time step d_i. We set $\epsilon = 1/2$. Then the *Online Reverse Auction* works as follows.

- $t = 0$: $S^1 = \emptyset$, $\rho = 1/2$, $b_1 = 1 \le \frac{V_1(S^1)}{\rho} = 1 \le \mathcal{B}^1 = 2$, thus $p_1 = 1, S = \{1\}$.

- $t = 2$: $S^1 = \{1\}, \rho = 1/2, b_2 = 2 > \frac{V_2(S^1)}{\rho} = 1$, thus $p_2 = 0$.
- $t = 4$: $S^1 = \{1\}, \rho = 1/2, b_3 = 3 > \frac{V_3(S^1)}{\rho} = 1$, thus $p_3 = 0$.
- $t = 6$: $S^2 = \emptyset$, $\rho = 1/2$, $b_4 = 1 \le \frac{V_4(S^2)}{\rho} = 1 \le \mathcal{B}^2 = 4$, thus $p_4 = 1, S = \{1, 4\}$.
- $t = 7$: $d^2 = t$ $S' = \{1, 2, 3\}$, $\mathcal{B}^1 = 2$, update $\rho = 1/4$. $b_4 = 1 \le \frac{V_4(S^2\backslash\{4\})}{\rho} = 2 \le \mathcal{B}^2 - p_4 + p_4 = 4$, and $\frac{V_4(S^2\backslash\{4\})}{\rho} = 2 > p_4 = 1$, thus increase p_4 to 2.

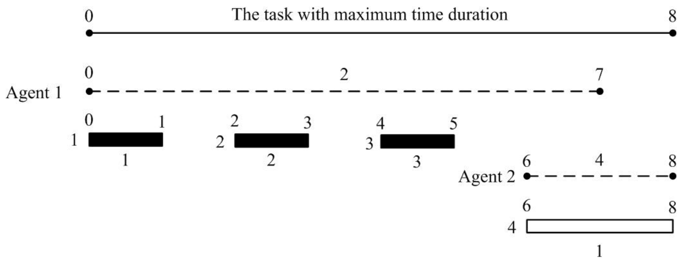


Fig. 3 An example illustrating how the *Online Reverse Auction* works, where the solid line represents the task with maximum time, the dotted lines represent the agent, the filled rectangles represent the social neighbors of agent 1, the hollow rectangle represents the social neighbor of agent 2. The 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.

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

IV. Mechanism Analysis

In the following, we present the theoretical analysis, demonstrating that *MTSC* can achieve the desired properties.

Lemma1. *MTSC is computationally efficient.*

Proof: It suffices to prove that both Algorithm 1 and Algorithm 2 are computationally efficient.

In Algorithm 1, computing the influence for all registered users (Lines 2-8) takes $O(n \cdot \max_{j\in J}|SN^j| \cdot m^2)$ 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 10-15) takes $O(n^2)$ time. Therefore, the running time of *Agent Selection* is bounded by $O(\max\{\max_{j\in J}|SN^j|nm^2, n^2\})$.

Note that the running time of Algorithm 1 is very conservative since the number of agents is much less than *n* in practice.

In Algorithm 2, 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 6-21) is dominated by finding the social neighbor with maximum marginal value (Line 11). The time complexity of computing the marginal value is $(\max_{j\in A}|SN^j| \cdot |\Gamma_i|)$, 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 6-21) takes $O(\max_{j\in A}|SN^j|m^2)$ time. Next, we analyze the time complexity of the function *DensityThreshold* (Algorithm 3). 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 3 is $O(|SN|m^2)$. Finally, according to the similar analysis, the time complexity of selecting new winners from all online social neighbors (Lines 24-34) is $O(\max_{j\in A}|SN^j|m^2)$. Hence, the running time of *Online Reverse Auction* is bounded by $O(|SN|m^2)$. ∎

Lemma 2. *MTSC is individually rational.*

Proof: From the lines 12-14 and lines 28-29 of Algorithm 2, we can see that $p_i \ge b_i$ if any social neighbor i is selected as a winner, otherwise $p_i = 0$. Therefore, we have $u_i \ge 0$. ∎

Lemma 3. *MTSC is budget feasible.*

Proof: MTSC allocates pro-rata budget $\mathcal{B}^j$ of total budget B to each agent $j \in A$ according to the influence (Line 17 of Algorithm 1). From the lines 12-14 and lines 28-29 of Algorithm 2, we can see that it is guaranteed that the current total payment does not exceed their budget $\mathcal{B}^j$. 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 $\mathcal{B}^j$. ∎

Lemma 4. *MTSC 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 $\hat{\theta}_i = (a_i, d_i, \Gamma_i, b_i, j)$. According to Algorithm 2, at each time step $t \in [a_i, d_i]$, there may be a new decision on whether to accept social neighbor *i*, and at what price. We use d_t^k, $\mathcal{B}_t^j$, ρ_t, and S_t^j 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 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 2, social neighbor *i* is paid for a price equal to the maximum price during $[a_i, d_i]$. Considering $ra_i \le a_i \le d_i \le rd_i$, 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 $t = ra_i$. In this case, there must be $c_i \le V_i(S_t^j)/\rho_t \le \mathcal{B}_t^j$, and $p_i = V_i(S_t^j)/\rho_t$. If i reports $b_i \le V_i(S_t^j)/\rho_t$, considering both $V_i(S_t^j)$ and $\mathcal{B}_t^j$ are independent of

b_i in this case, i wins still at time step $t = ra_i$ with same payment. If i reports $b_i > V_i(S_t^j)/\rho_t$, he will lose at time step $t = ra_i$, and $p_i = 0$.

Next, we consider the payment for i in time duration (t, d_t^k) (determined by lines 10-16 of Algorithm 2). For any time step $t' \in (t, d_t^k)$, considering the submodularity of $V(S)$, there must be $V_i(S_{t'}^j) \leq V_i(S_t^j)$. Note that the density threshold doesn't update in this case, i.e., $\rho_t = \rho_{t'}$. Therefore we have $p_i = V_i(S_{t'}^j)/\rho_{t'} \leq V_i(S_t^j)/\rho_t$ if i is selected at t'. Otherwise, $p_i = 0$. Therefore, a social neighbor cannot improve his payment by reporting false cost in time duration $[ra_i, d_t^k)$.

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

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 $t = ra_i$. Next, we consider the case when social neighbor i is not a winner by reporting the true cost at time step $t = ra_i$. In this case, there must be $c_i > V_i(S_t^j)/\rho_t$ or $V_i(S_t^j)/\rho_t > \mathcal{B}_t^j$. In case $c_i > V_i(S_t^j)/\rho_t$, if social neighbor i reports $b_i > V_i(S_t^j)/\rho_t$, then nothing changed. If social neighbor i reports $b_i \leq V_i(S_t^j)/\rho_t$, he would win with payment $V_i(S_t^j)/\rho_t$ at time step t . However, his utility will be negative. In addition, $\mathcal{B}_{t'}^j$ remains unchanged in both above cases, and thus social neighbor i's payment at time $t' > t$ is not affected. In case $V_i(S_t^j)/\rho_t > \mathcal{B}_t^j$, reporting a false cost does not affect the outcome at time step t or the residual budget $\mathcal{B}_{t'}^j$ at time step $t' > t$.

To sum up, reporting a false cost cannot improve social neighbor i's payment. ∎

The above four lemmas prove the following theorem.

Theorem 1. *MTSC is computationally efficient, individually rational, budget feasible and truthful (cost-truthful and time-truthful).*

V. PERFORMANCE EVALUATION

We have conducted thorough simulations to investigate the performance of *MSTC*. We implemented three benchmark mechanisms:

- **Approximate optimal (offline):** the approximate optimal offline solution with full knowledge about agents and social neighbors. For any agent $j \in A$, approximate optimal mechanism selects the winners from SN^j to maximize the value with budget $\mathcal{B}^j$. 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 $(1 - 1/e)$ approximation solution for each agent [25]. Note that the approximate optimal mechanism is untruthful.
- **Proportional share (offline):** an offline mechanism with full knowledge about agents and users. Different from the approximate optimal mechanism, it selects the winners using the proportional share rule [24] for each agent. The proportional share mechanism is truthful.
- **Random (online):** the online mechanism with dynamic agents and social neighbors. Different from *MTSC*, it selects the agents randomly.

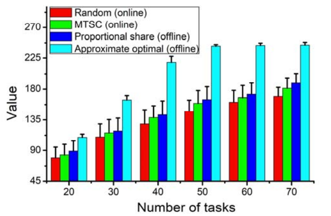
We first measure the value with different number of agents, number of tasks, budgets and initial density threshold (ϵ). Then we measure the running time of *MSTC* and verify the truthfulness of *MSTC*. All the simulations were run on a Windows 10 machine with Intel(R) Core(TM) i7-7560U CPU and 16 GB memory. Each measurement is averaged over 100 instances.

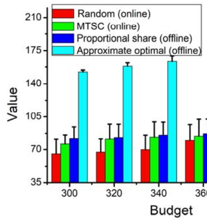
A. Simulation Setup

For our simulations, we use social circle data [26] 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. As the default setting, we choose 30 nodes from Facebook dataset as registered users and select 5 agents from the registered users with budget of 350. The arrival-departure interval of each agent is uniformly distributed over [1000, 2000] seconds. The arrival time of each agent is uniformly distributed over [0, 1000] seconds and the departure time is the arrival time add his/her arrival-departure interval. If the departure time of any agent is greater than 2000, we set the departure time of agent as 2000. Each task's end time is uniformly distributed over [500, 2000]. The arrival-departure interval of each social neighbor is uniformly distributed [100, 1000]. The arrival time of each social neighbor is uniformly distributed over [0, 500]. The cost of each bid is uniformly distributed over [3, 6]. The default number of tasks is 20. The value of each task is uniformly distributed over [5, 8]. The initial density threshold (ϵ) is 1. We will vary the value of key parameters to explore the impacts of these parameters.

B. Value

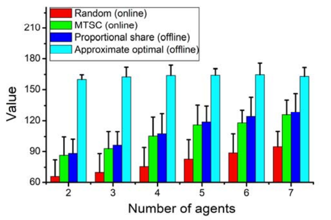
Fig. 4 compares the platform's value achieved by the *MTSC* mechanism against the three benchmarks. We can see that the platform obtains a higher value when the number of tasks increases or the budget constraint increases. The platform's value also increases with the number of agents since more social neighbors can bid for performing crowdsourcing tasks. The platform's value of online mechanisms decreases when the initial density thread increases. This is because when the initial density thread increases, the users are more difficult to be the winner. According to Algorithm 2, all winners should satisfy $\frac{V_i(S^j)}{\rho} \geq b_i$, i.e., $\frac{V_i(S^j)}{b_i} \geq \rho$. The offline mechanisms do not use the initial density thread as the parameter. The approximate optimal mechanism and the proportional share mechanism operate in the offline manner, where the platform has the full knowledge about agents and users. Thus, the two offline mechanisms always outperform the *MTSC* mechanism. It is shown that the proportional share mechanism sacrifices some performance to achieve the cost-truthfulness compared with the approximate optimal mechanism. The *MSTC* always achieves better performance than random mechanism since *MSTC* selects the agent delicately by considering both the period coverage and the user interests for the tasks.

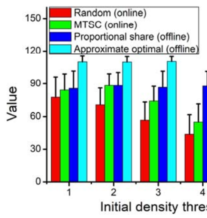




(a) Value versus number of tasks

(b) Value versus budget



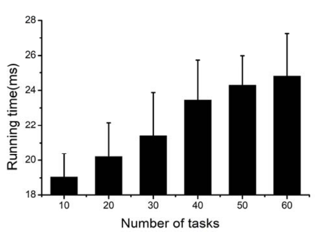


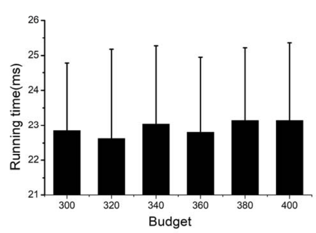
(c) Value versus number of agents

(d) Value versus initial density threshold (ϵ)

Fig.4 Value of the platform

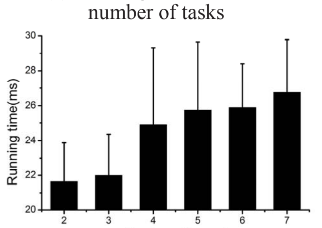
C. Running Time

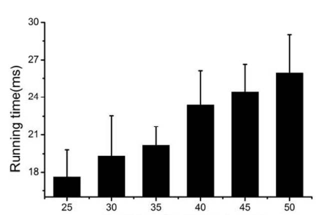




(a) Running time versus number of tasks

(b) Running time versus budget





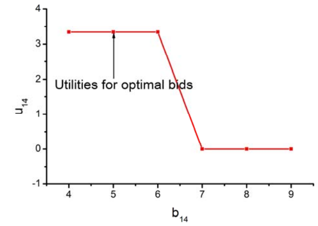
(c) Running time versus number of agents

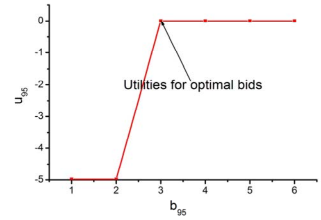
(d) Running time versus registered users

Fig. 5 Running time of *MSTC*.

Fig.5 shows the running time of *MSTC* with different number of tasks, budget, agents and registered users. It can be seen that the running time increase with the numbers of tasks, agents and registered users. When the number of agents increases, the number of social neighbors will increase accordingly. It leads to the increase of the running time. In fig.5 (b), the running time is stable when the budget increases. Thus, from Fig.5 we can infer that the running time increases with the number of the number of tasks, registered users and social neighbors, which is consistent with our analysis in Section IV.

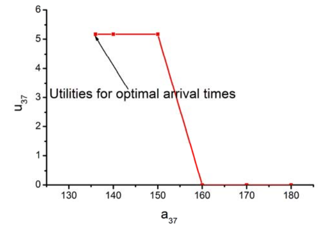
D. Truthfulness

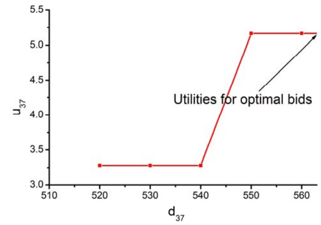




(a) $c_{14} = 5$

(b) $c_{95} = 3$





(c) $ra_{37} = 136$

(d) $rd_{37} = 563$

Fig. 6 Cost-truthfulness and time-truthfulness of *MTSC*.

We verified the cost-truthfulness of *MTSC* by randomly picking two social neighbors (ID=14 and ID=95) and allowing them to bid prices that are different from their true costs. We illustrate the results in Fig.6. We can see that social neighbor 14 achieves its optimal utility if he/she bids truthfully ($b_{14} = c_{14} = 5$) in Fig.6(a) and social neighbor 95 achieves its optimal utility if he/she bids truthfully ($b_{95} = c_{95} = 3$) in Fig. 6(b). Then we further verified the time-truthfulness of *MSTC* by randomly picking one social neighbor (ID=37) and allowing him to report his arrival/departure times that are different from his true arrival/departure times. As shown in Fig. 6(c) and Fig. 6(d), social neighbor 37 achieves its optimal utility if he/she reports its true arrival and departure times ($ra_{37} = a_{37} = 136, rd_{37} = d_{37} = 563$).

VI. Related Work

A. Offline Incentive Mechanisms

Aware of the paramount importance of stimulating worker participation, various incentive mechanisms have been proposed for MCS systems. Yang *et al.* [21] propose two different models for crowdsourcing: the platform-centric model and user-centric model. In the platform-centric model, the platform provides a fixed reward shared by participating users. In user-centric model, users have more control over the payment they will receive. Xu *et al.* design the incentive mechanisms, which consider the issue of stimulating the biased requesters in the competing crowdsourcing market [27]. Jin *et al.* [28] propose a novel integrated framework for multi-requester MCS systems, called CENTURION, consisting of a data aggregation mechanism and an incentive mechanism. The data aggregation mechanism of CENTURION takes into consideration workers' reliability, and calculates highly accurate aggregated results for the requesters. However, all of these studies do not consider the online arrival of users. In this paper, we aim to propose an online mechanism to select a subset of dynamic users before a specific deadline.

B. 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.* [8] propose *OMZ* and *OMG* models, which follow the multiple-stage sampling-accepting process. 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. Xiao *et al.* [9] propose an online task assignment for crowdsensing in predictable mobile social networks. The *LOTA* algorithm for the *MLM* 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. In this paper, we not only take dynamic users into consideration, but also take dynamic tasks and agents into consideration.

C. Incentive Mechanisms with Social Network

Social networks have been extensively discussed in mobile crowdsourcing [29, 30]. When the participants are not enough to complete the sensing tasks, the social network can help to spread the tasks to more potential participants. Wang *et al.* propose 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 [31]. Nguyen *et al.* propose the notions of node observability and coverage utility score, and design a new context-aware approximation algorithm to find vertex cover that is tailored for crowd-sensing tasks in opportunistic mobile social networks [13]. Overall, there is no off-the-shelf incentive mechanism designed in the literature, which takes advantage of social network to recruit more professional users to complete the tasks in the online manner.

VII. Conclusion

In this paper, we have presented a two-tiered social crowdsourcing architecture to solve the insufficient participation problem in online mobile crowdsourcing systems by enabling the selected registered users to recruit more users from their social circles. In our proposed system model, the tasks are associated with different end times, and both the registered users and social neighbors have different arrival/departure times. We have designed an online incentive mechanism consisting of *Agent Selection* step and *Online Reverse Auction* step. Through both rigorous theoretical analysis and extensive simulations, we have demonstrated that the proposed incentive mechanism achieves computational efficiency, individual rationality, budget feasibility, and truthfulness.

Acknowledgment

This work has been supported in part by the NSFC (No. 61472193, 61472192, 61502251), and NSF (No. 1444059, 1717315).

References

[1] Sensorly. Sensorly. http://www.sensorly.com.
[2] P. Mohan, V. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," in *Proc. 6th ACM SenSys*, Nov. 2008, pp. 323–336.
[3] A. Thiagarajan et al., "Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones," in *Proc. 7th ACM SenSys*, Nov. 2009, pp. 85–98.
[4] 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,* Apr. 2010, pp. 105–116.
[5] N. Maisonneuve, M. Stevens, M. Niessen, and L. Steels, "Noisetube: M easuring and mapping noise pollution with mobile phones," in *Proc. ITEE*, May 2009, pp. 215–228.
[6] J. Xu, J. Xiang, 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
[7] J. Xu, J. Fu, D. Yang, L. Xu, L. Wang, Tao Li. "FIMI:A Constant Frugal Incentive Mechanism for Time Window Coverage in Mobile Crowdsensing," *Journal of Computer Science and Technology*, 32 (5): 919-935, 2017.
[8] 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.
[9] M. Xiao et al., "Online Task Assignment for Crowdsensing in Predictable Mobile Social Networks," *IEEE Trans. Mobile Computing*, vol. 16, no. 8, 2017, pp. 2306–2320.
[10] https://www.analysys.cn/analysis/trade/detail/1000513/
[11] https://www.mturk.com/mturk/welcome
[12] https://www.kickstarter.com/
[13] P. Nguyen, and K. Nahrstedt. "Crowdsensing in Opportunistic Mobile Social Networks: A Context-aware and Human-centric Approach." *arXiv:1704.08598*, 2017.
[14] J. Xu, J. Xiang, Y. Li, et al. "Incentivize maximum continuous time interval coverage under budget constraint in mobile crowd sensing," *Wireless Networks*, vol. 23, no.5, pp.1549–1562, 2017.
[15] http://crowdsource.google.com/imagelabeler
[16] https://translate.google.com/community
[17] http://task.qq.com/
[18] http://test.baidu.com/crowdtest/community/index
[19] https://baike.baidu.com/
[20] https://www.stepes.com/
[21] 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.
[22] J. Li, Z. Cai, M. Yan, Y. Li, "Using crowdsourced data in location-based social networks to explore influence maximization". In: *Proc. IEEE INFOCOM*, 2016, pp. 1-9
[23] P. Jaccard, "The distribution of the flora in the alpine zone". *New phytologist*, 11(2), 37-50, 1912
[24] Y. Singer, "Budget feasible mechanisms," *in Proc. IEEE FOCS*, 2010, pp.765–774.
[25] S. Khullera, A. Mossb, and J. Naor, "The budgeted maximum coverage problem," *Information Processing Letters*, vol. 70, pp. 39–45, 1999.
[26] J. McAuley and J. Leskovec. "Learning to Discover Social Circles in Ego Networks". *NIPS*, 2012, pp. 539-547.
[27] J. Xu, H. Li, Y. Li, D. Yang, and T. Li, "Incentivizing the Biased Requesters: Truthful Task Assignment Mechanisms in Crowdsourcing", *in Proc. IEEE SECON*, 2017, pp. 1-9.
[28] H. Jin, L. Su, and K. Nahrstedt, "Centurion: Incentivizing multi-requester mobile crowd sensing," in *Proc. IEEE INFOCOM,* 2017. pp.1-9
[29] J. Xu, Z. Rao, L. Xu, D. Yang, T. Li. "Mobile Crowd Sensing via Online Communities: Incentive Mechanisms for Multiple Cooperative Tasks," in *Proc. IEEE MASS*, 2017, pp. 171-179.
[30] Y. Chen, B. Li, and Q. Zhang, "Incentivizing crowdsourcing systems with network effects," in *INFOCOM*, 2016.
[31] W. Wang and Y. Jiang, "Multiagent-Based Allocation of Complex Tasks in Social Networks," *IEEE Transactions on Emerging Topics in Computing*, 3(4):571 – 584, 2015.