Incentive Mechanisms for Multiple Cooperative Tasks with Compatible Users in Mobile Crowd Sensing via Online Communities

Jia Xu[1], Zhengqiang Rao[1], Lijie Xu[1], Dejun Yang[2], 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, 1216043122@njupt.edu.cn, ljxu@njupt.edu.cn, djyang@mines.edu, towerlee@njupt.edu.cn

***Abstract*—Mobile crowd sensing emerges as a new paradigm which takes advantage of the pervasive sensor-embedded smartphones to collect data efficiently. Many incentive mechanisms for mobile crowd sensing have been proposed. However, none of them takes into consideration the cooperative compatibility of users for multiple cooperative tasks. In this paper, we design truthful incentive mechanisms to minimize the social cost such that each of the cooperative tasks can be completed by a group of compatible users. We consider that the mobile crowd sensing is launched in an online community. We study two bid models and formulated the *Social Optimization Compatible User Selection* (*SOCUS*) problem for each model. We also define three compatibility models and use real-life relationships from social networks to model the compatibility relationships. We design two reverse auction based incentive mechanisms, *MCT-M* and *MCT-S*. Both of them consist of two steps: *compatible user grouping* and *reverse auction*. We further present a user grouping method through neural network model and clustering algorithm. Through both rigorous theoretical analysis and extensive simulations, we demonstrate that the proposed mechanisms achieve computational efficiency, individual rationality and truthfulness. In addition, *MCT-M* can output the optimal solution. By using neural network and clustering algorithm for user grouping, the proposed incentive mechanisms can reduce the social cost and overpayment ratio further with less grouping time.**

Keywords—Mobile crowd sensing; Incentive mechanism design; Online community; Compatibility

I. INTRODUCTION

Smartphones are widely available in the recent years. The worldwide smartphone market reached a total of 1.45 billion units shipped in 2016. From there, shipments will reach 1.71 billion units in 2020 [1]. Nowadays, smartphones are integrated with a variety of sensors such as camera, light sensor, GPS, accelerometer, digital compass, gyroscope, microphone, and proximity sensor. These sensors can collectively monitor a diverse range of human activities and surrounding environment. Mobile crowd sensing has become an efficient approach to meeting the demands in large-scale sensing applications [2], such as Sensorly [3] for 3G/WiFi discovery, TrMCD [4] for estimating user motion trajectory, crowd-participated system [5] for bus arrival time prediction, and participAct [6] for urban crowdsensing.

Incentive mechanisms are crucial to mobile crowd sensing while the smartphone users spend their time and consume battery, memory, computing power and data traffic of device to sense, store and transmit the data. Moreover, there are potential privacy threats [32, 33] to smartphone users by sharing their sensed data with location tags, interests or identities. A lot of research efforts have been focused on designing incentive mechanisms to entice users to participate in mobile crowd sensing system. However, they either focus on the multiple independent task scenario [7, 9-17], where each task only needs one user to perform, or pay attention to the single cooperative task scenario [8, 18], where the task requires a group of users to perform cooperatively. An incentive mechanism for multiple cooperative tasks has been designed in [19, 24], however, they neglect the relation among users.

The multiple cooperative task scenarios are very common. For example, the construct of fingerprint database [4] requires enough users to report sensor readings such that the correctness of trajectory can be guaranteed. In the bus arrival time prediction system [5], insufficient amount of uploaded information may result in inaccuracy in matching the bus route. Many time window dependent crowd sensing tasks [8], such as continuous measure of trace, traffic condition, noise and air pollution need a large sample space such that its result has statistical meaning. All above applications require users' collective contributions.

In multiple cooperative task scenarios, people would prefer to cooperate with trustworthy friends, especially when people are required to shall their privacy with the cooperators for performing sensing tasks. For example, the users in the bus arrival time prediction system [5] need to shall their location information with other users to guarantee that the pieces of sensed data from multiple users can be assembled to picture the intact bus route status. For the monitoring tasks [8], the users can allocate the sensing time intervals according to their private future schedules, habits, preferences or behavior profiles [20]. Furthermore, the cooperation of friends in mobile crowd sensing also helps to potential sensory data aggregation and corresponding local computation in the mobile devices in order to reduce network traffic and privacy threats. Thus choosing the compatible users to perform cooperative tasks can improve not only the participation willingness of users, but also the quality and success rates of mobile crowd sensing service.

In this paper, we consider that the mobile crowd sensing with multiple cooperative tasks is launched in an online community, in which the members (referred as users in the rest of this paper) are interested in participating sensing tasks. Each of cooperative tasks requires a specific amount of

This is an extended and enhanced version of the paper [31] that appeared in IEEE MASS 2017

compatible users to perform. We use real-life relationships from social networks to model the compatibility relationships. The objective is designing truthful incentive mechanisms to minimize the social cost (the total cost of winners) such that each cooperative task can be completed by a group of compatible users. In our system model, each user submits the tasks it can perform and corresponding bid prices. Meanwhile, each user can submit a set of recommended users according to its preference. Specifically, if there is no recommended user, the user can simply submit the empty set. The platform selects a subset of users and notifies winners of the determination. The winners perform the sensing tasks and send data back to the platform. Finally, each user obtains the payment, which is determined by the platform. The process is illustrated by Fig.1.

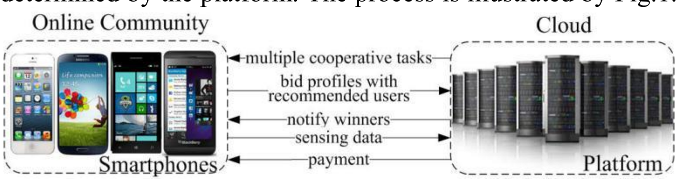


Fig. 1 Mobile crowd sensing process with multiple cooperative tasks

The problem of designing truthful incentive mechanisms to minimize the social cost for such mobile crowd sensing system is very challenging. First, the compatibility models should be defined to measure the different compatibility levels. Second, when selecting winners for tasks, the incentive mechanisms should consider not only the social optimization but also the compatibility of the users. Moreover, the user can take a strategic behavior by submitting dishonest recommended users or bid price to maximize its utility. Finally, the users may don't know the *compatible user set* exactly in some situations.

The main contributions of this paper are as follows:

- To the best of our knowledge, this is the first work to design truthful incentive mechanisms for the mobile crowd sensing system, where each task needs to be performed by a group of compatible users.
- We present two bid models, and formulate the *Social Optimization Compatible User Selection* (*SOCUS*) problem for each. We further present three compatibility models, which can depict the different compatibility levels, and use real-life relationships from social networks to model the compatibility relationships.
- We design two incentive mechanisms *MCT-M* and *MCT-S* for two bid models. We show that the designed mechanisms satisfy desirable properties of computational efficiency, individual rationality and truthfulness. In addition, *MCT-M* can output the optimal solution.
- We introduce neural network method to learn the similarity between users, and group the users using clustering algorithm according to this similarity for the situations, where the users don't know the *compatible user set* exactly. Such user grouping method also helps to reduce the social cost and overpayment ratio further.

The rest of the paper is organized as follows. Section Ⅱ formulates two bid models and three compatibility models, and lists some desirable properties. Section Ⅲ presents the benchmark mechanisms for both *MCT-M* and *MCT-S*. Section IV and Section V present the detailed design and analysis of our incentive mechanisms for two bid models, respectively. Section Ⅵ presents user grouping method based on neural network and clustering algorithm. Performance evaluation is presented in Section Ⅶ. We review the state-of-art research in Section Ⅷ, and conclude this paper in Section Ⅸ.

II. System Model And Desirable Properties

In this section, we model the mobile crowd sensing system as a reverse auction and present two different bid models: multi-bid model and single-bid model. In the multi-bid model, each user can submit multiple task-bid pairs and can be recruited to work on a portion of submitted tasks. The single-bid model allows each user to bid a global price for multiple tasks it can perform. Each user is required to perform all submitted tasks once he is selected as a winner in the single-bid model. Thus the single-bid model is suitable for the single-minded users, while multi-bid model provides more flexibility to the users. Moreover, we present three compatibility models of users: weak compatibility model, medium compatibility model and strong compatibility model. At the end of this section, we present some desirable properties.

A. Multi-bid Model

We consider a mobile crowd sensing system consisting of a social network application platform and an online community with many smartphone users. The platform resides in the cloud. The platform publicizes a set $T=\{t_1,t_2,\dots,t_m\}$ of m cooperative tasks in an online community $U=\{1,2,\dots,n\}$ of n smartphone users, who are interested in participating sensing tasks. Each task $t_j\in T$ is associated with the *cooperative index* r_j, which is the least number of compatible users to perform t_j.

Each user i submits a 2-tuple $B_i=(\beta_i,\zeta_i)$, where $\beta_i=\{\beta_i^1,\beta_i^2,\dots,\beta_i^{k_i}\}$ is a set of k_i task-bid pairs. The task-bid pair for task j is denoted by $\beta_i^j=(t_i^j,b_i^j)$, $t_i^j\in T$. Each t_i^j is associated with the cost c_i^j, which is the private information and known only to user i. b_i^j is the claimed cost, which is the bid price that user i wants to charge for performing t_i^j. Each user can submit a set of recommended users, called *compatible user set*, according to its preference. The user prefers to cooperate with the users in its *compatible user set* to perform the tasks. We also consider that the real *compatible user set* is the private information and known only to user i. $\zeta_i\subseteq U$ is the claimed *compatible user set* of i.

Given the task set T and the bid profile $\boldsymbol{B}=(B_1,B_2,\dots,B_n)$, the platform calculates the winning task-bid pair set $\beta_S\subseteq\cup_{i\in U}\beta_i$ and the payment p_i^j for each winning task-bid pair $\beta_i^j\in\beta_S$. The payment for each winner i is $p_i=\sum_{\beta_i^j\in\beta_i\cap\beta_S}p_i^j$. A user i is called a winner and added into winner set S if it has at least one winning task-bid pair, i.e., $\beta_i\cap\beta_S\neq\emptyset$. We define the utility of user i as the difference between the payment and its real cost:

$$u_i=p_i-\sum_{\beta_i^j\in\beta_i\cap\beta_S}c_i^j \quad (1)$$

Since we consider the users are selfish and rational individuals, each user can behave strategically by submitting a dishonest *compatible user set* or dishonest bid prices to maximize its utility. We assume that the truthfulness of submitted task can be achieved since they can be verified by the platform. In order to prevent the monopoly and guarantee the sensing quality, we assume each cooperative task can be completed by at least two different groups of compatible users. Here, we say two groups are different iff there is at least one different user between them. This assumption is reasonable for mobile crowd sensing systems as made in [7, 8, 9]. If a task can only be completed by the unique group of compatible users, the platform can simply remove it from *T*.

The incentive mechanism $\mathcal{M}(T, B)$ outputs a winning task-bid pair set β_S and a payment profile $\mathbf{p} = (p_1, p_2, \dots, p_n)$. The objective is minimizing the social cost such that each of cooperative tasks in *T* can be completed by a group of compatible users. We will present the compatibility models in Section Ⅱ-*C*. We refer this problem as *Social Optimization Compatible User Selection* (*SOCUS*) problem, which can be formulated as follows:

$$\min \sum\nolimits_{\beta_i^j \in \beta_i \cap \beta_S} c_i^j$$

$$\text{s.t.} \ \sum\nolimits_{\beta_i^j \in \beta_i \cap \beta_S, t_j = t_i^j} |\beta_i^j| \geq r_j, \forall t_j \in T$$

B. Single-bid Model

The definitions of $T,\ U,\ \zeta_i,\ t_i,\ r_j, t_i^j$ are the same as those in Section Ⅱ-*A*. Each user *i* submits a 3-tuple $B_i = (\beta_i, b_i, \zeta_i)$, where $\beta_i = \{t_i^1, t_i^2, \dots, t_i^{k_i}\}$ is a set of k_i tasks. The task set β_i is associated with the cost c_i, which is the private information and known only to user *i*. b_i is the claimed cost. We also consider the real *compatible user set* is the private information and known only to user *i*.

Given the task set *T* and the bid profile $\boldsymbol{B} = (B_1, B_2, \dots, B_n)$, the platform calculates the winner set $S \subseteq U$ and the payment p_i for each winner $i \in S$. We define the utility of user *i* as:

$$u_i = p_i - c_i \qquad (2)$$

A user can behave strategically by submitting a dishonest *compatible user set* or a dishonest bid price to maximize its utility. The incentive mechanism $\mathcal{M}(T, B)$ outputs a winner set *S* and a payment profile $\mathbf{p} = (p_1, p_2, \dots, p_n)$. The objective is minimizing the social cost such that each of the cooperative tasks in *T* can be completed by a group of compatible users. The *Social Optimization Compatible User Selection* (*SOCUS*) problem in the single-bid model can be formulated as follows:

$$\min \sum\nolimits_{i \in S} c_i$$

$$\text{s.t.} \ \sum\nolimits_{i \in S, t_i^j \in \beta_i, t_j = t_i^j} |t_i^j| \geq r_j, \forall t_j \in T$$

C. Compatibility Model

In this subsection, we present three compatibility models to depict the different compatibility levels:

- **Weak Compatibility Model:** The two users *i* and *j* satisfy the weak compatibility (denote as $i \doteq j$) if $j \in \zeta_i$ or $i \in \zeta_j$ for any $i, j \in U$. We consider that the relation of weak compatibility is symmetric and transitive. Then we define *Weak Compatibility Group* (*WCG*) as $\{i | i \doteq j, \forall i, j \in U\}$. Essentially, the weak compatibility model is established on the one-way preferences between the users.
- **Medium Compatibility Model:** We define the transitive relation $\rhd$: If $k \in \zeta_i \wedge j \in \zeta_k, \forall i, j, k \in U$, we say $i \rhd j$. The two users *i* and *j* satisfy the medium compatibility (denote as $i \triangleq j$) if $i \rhd j$ and $j \rhd i$ for any $i, j \in U$. Then we define *Medium Compatibility Group* (*MCG*) as $\{i | i \triangleq j, \forall i, j \in U\}$. The medium compatibility model is established on the transitive two-way preferences between the users.
- **Strong Compatibility Model:** The two users *i* and *j* satisfy the strong compatibility (denote as $i \hat{=} j$) if $j \in \zeta_i$ and $i \in \zeta_j$ for any $i, j \in U$. We consider that the relation of strong compatibility is symmetric and transitive. Then we define *Strong Compatibility Group* (*SCG*) as $\{i | i \hat{=} j, \forall i, j \in U\}$. The strong compatibility model is established on the two-way preferences between the users.

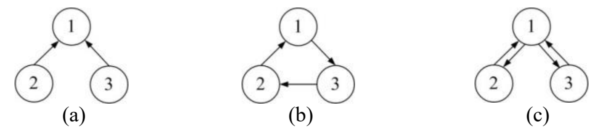


Fig. 2 Examples illustrating *WCG*, *MCG* and *SCG* with 3 users, where the disks represent users, and the arrows represent the *compatible user sets*: (a) An example of *WSG*. (b) An example of *MSG*. (c) An example of *SCG*.

Obviously, the medium compatibility model is a special case of strong compatibility model, and the weak compatibility model is a special case of medium compatibility model. We give three simple examples for illustrating *WCG*, *MCG* and *SCG* in Fig.2, respectively.

D. Desirable Properties

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

- **Computational Efficiency:** An incentive mechanism $\mathcal{M}$ is computationally efficient if the outcome can be computed in polynomial time.
- **Individual Rationality:** Each user will have a non-negative utility when bidding its true cost and *compatible user set*, i.e., $u_i \geq 0, \forall i \in U$.
- **Truthfulness:** An incentive mechanism is compatibility- and cost-truthful (called truthful simply) if reporting the true *compatible user set* and cost is a weakly dominant strategy for all users. In other words, no user can improve its utility by submitting a false *compatible user set* or cost, no matter what others submit.
- **Social Optimization:** A mechanism achieves social optimization if it can output the optimal solution.

III. Benchmark Mechanisms

In this section, we consider the special cases for multi-bid model and single-bid model, respectively, where any user *i* submit the claimed *compatible user set* $\zeta_i = U / \{i\}$. This means any user can cooperate with all other users without considering the compatibility of users. We present the incentive mechanisms for multiple cooperative tasks in these special cases, and treat them as the benchmark mechanisms for *MCT-M* and *MCT-S*, respectively.

A. Benchmark Mechanism for the Multi-bid Model

The *Benchmark Mechanism for the Multi-bid Model* (*Benchmark-M*) consisting of user selection phase and payment determination phase. Let U^j and S^j be the users bidding for any task t_j and the winners of any task t_j, respectively. In winner selection phase, we propose an optimal algorithm, which selects r_j task-bid pairs with minimum total bid price as the winning task-bid pairs for any task $t_j \in T$. Obviously, this can output the optimal solution for the *SOCUS* problem. In payment determination phase, we compute payment based on the *VCG* payment rule [10]: A winning task-bid pair will be paid an amount equal to the benefit it introduces to the system, i.e., the difference between other task-bid pairs' minimum social cost with and without it (Line 13 of Algorithm 1), where function $cost()$ means the minimum social cost computed by selection phase. The whole process is illustrated in Algorithm 1.

For *Benchmark-M*, we have the following theorem.

Theorem 1. *Benchmark-M is computationally efficient, individually rational, cost-truthful and an optimal algorithm of SOCUS problem in the special case of multi-bid model.*

Proof: The running time of obtain U_j for any task t_j (Line3) takes $O(mn)$ since there at most *mn* task-bid pairs. The while-loop (Line4-9) is dominated by sorting the task-bid pairs based on bid price (Line5), which takes $O(nlogn)$ since there are at most *n* task-bid pairs for each task. There are *m* tasks, and the winner selection phase takes $O(m\max\{mn, nlogn\})$ time. In the payment determination phase, a process similar to winner selection phase is executed for each winning task-bid pair. Since there are at most $\sum_{j=1}^{m} r_j$ winning task-bid pairs, running time of the Algorithm 1 is bounded by $O((\sum_{j=1}^{m} r_j)m\max\{mn, nlogn\})$.

It is easy to know that *benchmark-M* can output the optimal solution of *SOCUS* problem. Since we adopt *VCG* payment rule [10], which is known as an individually rational and cost-truthful auction, *benchmark-M* is individually rational and cost-truthful. ■

Algorithm 1: Benchmark-M

Input: task set T, bid profile $\boldsymbol{B}$, user set U
// Winner Selection Phase
1 $S \leftarrow \emptyset$; $\beta_S \leftarrow \emptyset$; $cost \leftarrow 0$;
2 **foreach** $t_j \in T$ **do**
3 $\quad S^j \leftarrow \emptyset$; $U^j \leftarrow \{i | i \in U, t_j = t_i^j\}$;
4 $\quad$ **while** $|S^j| < r_j$ **then**
5 $\quad\quad i \leftarrow \arg min_{i \in U^j / S^j} b_i^j$;
6 $\quad\quad \beta_S \leftarrow \beta_S \cup \{\beta_{i'}^j\}$;
7 $\quad\quad S^j \leftarrow S^j \cup \{i'\}$; $S \leftarrow S \cup \{i'\}$;
8 $\quad\quad cost \leftarrow cost + b_{i'}^j$;
9 $\quad$ **end**
10 **end**
//Payment Determination Phase
11 **foreach** $i \in U$ **do** $p_i \leftarrow 0$;
12 **foreach** $\beta_i^j \in \beta_S$ **do**
13 $\quad p_i^j \leftarrow cost(\cup_{i \in U} \beta_i \setminus \{\beta_i^j\}) - (cost(\cup_{i \in U} \beta_i) - b_i^j)$;
14 **end**
15 **foreach** $i \in S$ **do** $p_i = \sum_{\beta_i^j \in \beta_i \cap \beta_S} p_i^j$;
16 **return** $(cost, \beta_S, \mathbf{p})$;

B. Benchmark Mechanism for the Single-bid Model

First of all, we attempt to find an optimal algorithm for the *SOCUS* problem in the special case of single-bid model. Unfortunately, as the following theorem shows, this problem is NP-hard.

Theorem 2. *The SOCUS problem in the special case of single-bid model is NP-hard.*

Proof: We demonstrate that the *SOCUS* problem in the special case of single-bid model belongs to NP firstly. Given an instance of *SOCUS* problem in the special case of single-bid model, we can check whether the winners can perform all tasks and check whether the social cost is at most *k*. This process can be end up in polynomial time.

Next, we prove the *SOCUS* problem in the special case of single-bid model is NP-hard by giving a polynomial time reduction from the NP-hard *Weighted Set Multiple Cover* problem, *WSMC*.

Instance of *WSMC* (denoted by *A*): For an universe set $\mathcal{T} = \{\mathcal{t}_1, \mathcal{t}_2, \dots, \mathcal{t}_m\}$ of *m* elements, each $\mathcal{t}_j$ is associated with a positive integer $\mathcal{r}_j$, for $j \in \{1,2,\dots,m\}$. There is a family of sets $G = \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_n\}$ and a positive real k, each $\mathcal{T}_i \subseteq \mathcal{T}$ has its weight c'_i for $i \in \{1,\dots,n\}$. The question is whether exists a set $G' \subseteq G$ with $\sum_{\mathcal{T}_i \in G'} c'_i \leq k$, such that any element $\mathcal{t}_j \in \mathcal{T}$ can be covered by $\mathcal{r}_j$ times?

We consider a corresponding instance of *SOCUS* problem in the special case of single-bid model (denoted by *B*): There is an universe task set $T = \{t_1, t_2, \dots, t_m\}$ of *m* tasks, and each task t_j is associated with a task threshold r_j, for $j \in \{1,2,\dots,m\}$, where r_j is a positive integer. There is a family of task sets $E = \{\beta_1, \beta_2, \dots, \beta_n\}$ and a positive real k, each user $i \in U$ is associated with a task set β_i and a cost c_i for $i \in \{1,\dots,n\}$. The question is whether exists a set $E' \subseteq E$ with $\sum_{\beta_i \in E'} c_i \leq k$, such that any task $t_j \in T$ can be performed by r_j times?

This reduction from *A* to *B* ends in polynomial time. We can simply see that *x* is a solution of *A* if and only if *x* is a solution of *B*. ■

Algorithm 2: Benchmark-S

Input: task set T, bid profile $\boldsymbol{B}$, user set U
//Winner Selection Phase
$T' \leftarrow T$, $S \leftarrow \emptyset$;
foreach $t_j \in T'$ **do** $r'_j \leftarrow r_j$;
while $T' \neq \emptyset$ **do**
$\quad i \leftarrow \arg min_{k \in U \setminus S} \frac{b_k}{|T' \cap T_k|}$;
$\quad S \leftarrow S \cup \{i\}$;
$\quad$ **foreach** $t_j \in T' \cap T_i$ **do**
$\quad\quad r'_j \leftarrow r'_j - 1$;
$\quad\quad$ **if** $r'_j = 0$ **then** $T' \leftarrow T' \setminus \{t_j\}$;
$\quad$ **end for**
end while
//Payment Determination Phase
foreach $i \in U$ **do** $p_i \leftarrow 0$;
foreach $i \in S$ **do**

13 $U' \leftarrow U\backslash\{i\}, T'' \leftarrow T, S' \leftarrow \emptyset$;
14 **foreach** $t_j \in T''$ **do** $r'_j \leftarrow r_j$;
15 **while** $T'' \neq \emptyset$ **do**
16 $i_k \leftarrow \arg min_{k \in U' \backslash S'} \frac{b_k}{|T'' \cap T_k|}$;
17 $p_i \leftarrow \max\{p_i, \frac{|T'' \cap T_i|}{|T'' \cap T_{i_k}|} b_{i_k}\}$;
18 $S' \leftarrow S' \cup \{i_k\}$;
19 **foreach** $t_j \in T'' \cap T_{i_k}$ **do**
20 $r'_j \leftarrow r'_j - 1$;
21 **if** $r'_j = 0$ **then** $T'' \leftarrow T''\backslash\{t_j\}$;
22 **end for**
23 **end while**
24 **end for**

Since the *SOCUS* problem in the special case of single-bid model is NP-hard, it is impossible to compute the winner set with minimum social cost in polynomial time unless P=NP. We design *Benchmark Mechanism for the Single-bid Model* (*Benchmark-S*) through a greedy approach. Illustrated in Algorithm 2, the reverse auction consists of winner selection phase and payment determination phase.

In the winner selection phase, the users are essential sorted according to the *Effective Unit Cost*. Given any uncovered task set T', the *Effective Unit Cost* of user *i* is defined as $\frac{b_i}{|T' \cap T_i|}$. In each iteration of the winner selection phase, we select the user with minimum *Effective Unit Cost* over the unselected user set $U\backslash S$ as the winner until the winners together can perform each task $t_j \in T$ by r_j times.

In payment determination phase, for each winner $i \in S$, we execute the winner selection phase over $U\backslash\{i\}$, and the winner set is denoted as S'. We compute the maximum price that the user *i* can be selected instead of each user in S'.

Next, we present the theoretical analysis of *benchmark-S*.

Lemma 1. *Benchmark-S is computationally efficient.*

Proof: Finding the user with minimum with minimum *Effective Unit Cost* takes $O(mn)$, where computing the value of $|T' \cap T_k|$ takes $O(m)$. Hence, the while-loop (Line3-10) takes $O(mn^2)$. In each iteration of the for-loop (line 12-23), a process similar to line 3-10 is executed. Hence the time complexity of the whole auction is dominated by this for-loop, which is bounded by $O(mn^3)$. ■

Lemma 2. *Benchmark-S is individually rational.*

Proof: Let i_k be user *i*'s replacement which appears in the *i*th place in the sorting over $U\backslash\{i\}$. Since i_k would not be at *i*th place if *i* is considered, we have $\frac{b_i}{|T' \cap T_i|} \leq \frac{b_{i_k}}{|T' \cap T_{i_k}|}$. Hence $b_i \leq \frac{|T' \cap T_i|}{|T' \cap T_{i_k}|} b_{i_k} = \frac{|T'' \cap T_i|}{|T'' \cap T_{i_k}|} b_{i_k}$, where the equality relies on the observation that $T' = T''$ for every *k*≤*i*, which is due to the fact that $S = S'$ for every *k*≤*i*. This is sufficient to guarantee $b_i \leq max_{k \in U' \backslash S'} \frac{|T'' \cap T_i|}{|T'' \cap T_{i_k}|} b_{i_k} = p_i$ ■

Before analyzing the truthfulness of *Benchmark-S*, we firstly introduce the Myerson's Theorem [13].

Theorem 3. *([14, Theorem 2.1]) An auction mechanism is truthful if and only if:*

- *The selection rule is monotone: If user i wins the auction by bidding b_i, it also wins by bidding $b_i \leq b_i$;*
- *Each winner is paid the critical value: User i would not win the auction if it bids higher than this value.*

Lemma 3. *Benchmark-S is truthful.*

Proof: Based on Theorem 3, it suffices to prove that the selection rule of *Benchmark-S* is monotone and the payment p_i for each influenced user *i* is the critical value. The monotonicity of the selection rule is obvious as bidding a smaller value cannot push influenced user *i* backwards in the sorting.

We next show that p_i is the critical value for the user *i* that bidding higher p_i could prevent *i* from winning the auction. Note $p_i = max_{k \in \{1,2,\ldots,L\}} \frac{|T'' \cap T_i|}{|T'' \cap T_{i_k}|} b_{i_k}$. If the user *i* bids $b_i > p_i$, he will be placed after *L* since $b_i > \frac{|T'' \cap T_i|}{|T'' \cap T_{i_L}|} b_{i_L}$ implies $\frac{b_i}{|T'' \cap T_i|} \geq \frac{b_{i_L}}{|T'' \cap T_{i_L}|}$. Hence, the user *i* would not win the action because the first *L* users have finished all tasks. ■

Lemma 4. *Benchmark-S can approximate the optimal solution within a factor of H_m, where $H_m = \sum_{i=1}^{m} \frac{1}{i} \leq \text{In} m + 1$.*

Proof: Since *SOCUS* problem in the special case of single-bid model is be equivalent to the *WSMC* problem, we can obtain the approximation ratio of H_m using the dual fitting method [22] for the *WSMC* problem. ■

As a conclusion of lemma 1 to lemma 4, we have the following theorem.

Theorem 4. *Benchmark-S is computationally efficient, individually rational, truthful, and H_m approximate for the special case of single-bid model.*

IV. Incentive Mechanism For The Multi-bid Model

In this section, we take the compatibility among users into consideration, and present an incentive mechanism for *Multiple Cooperative Tasks in the Multi-bid model* (*MCT-M*). *MCT-M* consists of two steps: *compatible user grouping* and *reverse auction*. *MCT-M* first divides the users into *compatible user groups*, in which each user is compatible with others. Afterwards, *MCT-M* performs a reverse auction mechanism to select the winning task-bid pairs and determine the payment for each user.

A. Compatible User Grouping

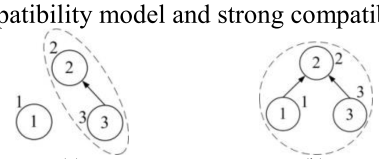
MCT-M first divides the users into *compatible user groups* based on the compatibility models defined in Section Ⅱ-*C*, and constructs *WCG*s, *MCG*s or *SCG*s.

For the weak compatibility model, we construct an undirected graph to represent the user compatibility relation based on the claimed *compatible user set*. For any $i, j \in U, i \neq j$, if there is $j \in \zeta_i$ or $i \in \zeta_j$, we add an edge between *i* and *j*. Then the *WCG*s can be constructed within $O(n^2)$ time through computing the connected components of the graph.

For the medium compatibility model, we construct a directed graph. For any $i, j \in U, i \neq j$, if there is $j \in \zeta_i$, we add a directed edge from i to j. Then we can construct *MCG*s through computing the strongly connected components of the graph, which can be solved within $O(n^2)$ time.

For the strong compatibility model, we construct an undirected graph. For arbitrary $i, j \in U, i \neq j$, if there is $j \in \zeta_i$ and $i \in \zeta_j$, we add an edge between i and j. Then we can construct *SCG*s through computing the connected components.

It is straightforward to construct the compatible user groups according to the original *compatible user sets*. However, the outcome of *compatible user grouping* depends strongly on the claimed *compatible user sets*. In other words, the users can change the outcome of grouping by misreporting their *compatible user sets*. We use the example in Fig.3 to illustrate that grouping according to the original *compatible user set* leads untruthfulness in weak compatibility model. Let $T = \{t_j\}$, $r_j = 2$, $U = \{1,2,3\}$, $\zeta_3 = \{2\}$, $\zeta_1 = \zeta_2 = \emptyset$, $b_1 = 1$, $b_2 = 2$, $b_3 = 3$. All users bid for task *j*. We first consider the case where all three users submit real *compatible user sets*. Obviously, $WCG = \{2,3\}$, $u_1 = 0$ since user 1 cannot cooperate with any user. We now consider the case where user 1 lies by submitting $\zeta_1 = \{2\}$. In this case, $WCG = \{1,2,3\}$, $S = \{1,2\}$ and the payment for user 1 would be 3 if we use *VCG* payment rule [10]. Thus, $u_1 = 3 - 1 = 2$. Note that user 1 improves its utility from 0 to 2 by lying about its *compatible user set*. The similar examples can be illustrated for both medium compatibility model and strong compatibility model.



(a) (b)

Fig. 3 An example showing the untruthfulness of grouping according to the original *compatible user sets* in the weak compatibility model, where the disks represent users, and the arrows represent the *compatible user sets*. The numbers beside the disks represent the cost for performing task *j*. The dotted disks represent *WCG*s: (a) All users submit real *compatible user sets.* (b) User 1 lies by submitting $\zeta_1 = \{2\}$.

Algorithm 3: Compatible User Grouping

Input: graph G
$A \leftarrow \emptyset$; $S_{\mathscr{k}} \leftarrow \emptyset$;
foreach $i \in \{1,2,\dots,\mathscr{m}\}$ **do**
  $U_i \leftarrow \emptyset$; $A \leftarrow A \cup U_i$;
end
Assign each user independently and uniformly at random to one of $\mathscr{m}$ subsets $U_1, U_2, \dots, U_{\mathscr{m}}$;
Let $A_{\mathscr{k}} \subseteq A$ be a random subset with size $\mathscr{k} - \mathscr{m}\lfloor \mathscr{k}/\mathscr{m} \rfloor$;
foreach $U_i \in A$ **do**
  if $U_i \in A_{\mathscr{k}}$ **then**
    if $|U_i| < \lceil \mathscr{k}/\mathscr{m} \rceil$ **then**
      $S_{\mathscr{k}} \leftarrow S_{\mathscr{k}} \cup U_i$; $U_i \leftarrow \emptyset$;
    else
      Let $U_i' \subseteq U_i$ be the set of $\lceil \mathscr{k}/\mathscr{m} \rceil$ users with highest indegree based only on edges from $U \backslash U_i$;
      $S_{\mathscr{k}} \leftarrow S_{\mathscr{k}} \cup U_i'$; $U_i \leftarrow U_i \backslash U_i'$;
    end
  else
    if $|U_i| < \lfloor \mathscr{k}/\mathscr{m} \rfloor$ **then**
      $S_{\mathscr{k}} \leftarrow S_{\mathscr{k}} \cup U_i$; $U_i \leftarrow \emptyset$;
    else
      Let $U_i' \subseteq U_i$ be the set of $\lfloor \mathscr{k}/\mathscr{m} \rfloor$ users with highest indegree based only on edges from $U \backslash U_i$;
      $S_{\mathscr{k}} \leftarrow S_{\mathscr{k}} \cup U_i'$; $U_i \leftarrow U_i \backslash U_i'$;
    end
  end
end
if $|S_{\mathscr{k}}| < \mathscr{k}$ then
  for i=1 **to** $\mathscr{k} - |S_{\mathscr{k}}|$ **do**
    Select j uniformly from $U \backslash S_{\mathscr{k}}$;
    $S_{\mathscr{k}} \leftarrow S_{\mathscr{k}} \cup \{j\}$;
  end
end
Group the users in $S_{\mathscr{k}}$ based on the specific compatibility model. Let $\mathcal{G}$ be the set of compatible users groups.
return $\mathcal{G}$;

To solve this issue, we introduce the *Random $\mathscr{m}$-Partition Mechanism* ($\mathscr{m}$*-RP*) [11], which is a randomized truthful mechanism for the approval voting [12]. We construct a directed graph G without self-loops: For any $i, j \in U, i \neq j$, if there is $j \in \zeta_i$, we add a directed edge from i to j. Then we select a subset $S_{\mathscr{k}}$ of $\mathscr{k}$ users to maximize the total indegree of selected users. In our system model, we adopt $\mathscr{m}$*-RP* to select $\mathscr{k}$ users with the maximum recommendations. Then *MCT-M* constructs *WCG*s, *MCG*s or *SCG*s based on the recommendations of $\mathscr{k}$ users selected through $\mathscr{m}$-RP.

The whole process of compatible user grouping is illustrated in Algorithm 3, which works as follows:

(1) The users are assigned independently and uniformly at random to one of $\mathscr{m}$ subsets (denoted as $U_1, U_2, \dots, U_{\mathscr{m}}$). Let A be the set of these $\mathscr{m}$ subsets.

(2) Select $\mathscr{k} - \mathscr{m}\lfloor \mathscr{k}/\mathscr{m} \rfloor$ subsets from A randomly. Let $A_{\mathscr{k}}$ be the set of these $\mathscr{k} - \mathscr{m}\lfloor \mathscr{k}/\mathscr{m} \rfloor$ subsets.

(3) For each $U_i \in A$, if $U_i \in A_{\mathscr{k}}$, select $\lceil \mathscr{k}/\mathscr{m} \rceil$ users from U_i with highest indegree based on edges from $U \backslash U_i$; if $U_i \notin A_{\mathscr{k}}$, select $\lfloor \mathscr{k}/\mathscr{m} \rfloor$ users from U_i with highest indegree based on edges from $U \backslash U_i$.

(4) If any subset U_i is smaller than the number of users needs to be selected, select all users in this subset.

(5) If the size of winner set $S_{\mathscr{k}}$ is smaller than $\mathscr{k}$, select $\mathscr{k} - |S_{\mathscr{k}}|$ additional users from the unselected users uniformly.

(6) Group the users in $S_{\mathscr{k}}$ based on the specific compatibility model.

B. Auction mechanism design

Algorithm 4: Reverse Auction for Multi-bid Model

Input: task set T, bid profile $\boldsymbol{B}$, compatible user group set $\mathcal{G}$, the set of $\mathscr{k}$ users $U_{\mathcal{G}}$
// Winner Selection phase
$S \leftarrow \emptyset$; $cost \leftarrow 0$; $\beta_S \leftarrow \emptyset$; $\mathcal{G} \leftarrow \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_d\}$;
for $k \leftarrow 1$ to d **do** $S_k \leftarrow \emptyset$;
foreach $t_j \in T$ **do**
  foreach $k \leftarrow 1$ **to** d **do**
    $S_k \leftarrow \emptyset$;
    $num_k \leftarrow |\{\beta_i^j | i \in \mathcal{G}_k, t_j = t_i^j\}|$;
    if $num_k \geq r_j$ **then**
      do

9 $i' \leftarrow \text{argmin}_{i \in \mathcal{G}_k \setminus S_k} b_i^j$;
10 $S_k \leftarrow S_k \cup \{i'\}$;
11 **until** $|S_k| \geq r_j$;
12 **end**
13 **end**
14 $k' \leftarrow \text{argmin}_{k \in \{1,2,\ldots,d\}} \sum_{i \in S_k, S_k \neq \emptyset} b_i^j$;
15 **foreach** $i \in S_{k'}$ **do** $\beta_S \leftarrow \beta_S \cup \{\beta_i^j\}$;
16 $cost \leftarrow cost + \sum_{i \in S_{k'}} b_i^j$;
17 $S \leftarrow S \cup S_{k'}$;
18 **end**
//Payment Determination Phase
19 **foreach** $i \in U$ **do** $p_i \leftarrow 0$;
20 **foreach** $\beta_i^j \in \beta_S$ **do**
21 $p_i^j \leftarrow cost(\cup_{i \in U_{\mathcal{G}}} \beta_i \setminus \{\beta_i^j\}) - (cost(\cup_{i \in U_{\mathcal{G}}} \beta_i) - b_i^j)$;
22 **end**
23 **foreach** $i \in S$ **do** $p_i = \sum_{\beta_i^j \in \beta_i \cap \beta_S} p_i^j$;
24 **return** $(cost, \beta_S, \mathbf{p})$;

Consider that the outcome of *compatible user grouping* is a set of d *compatible user groups* $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_d\}$. Let $U_{\mathcal{G}}$ be the set of $\mathcal{k}$ users. *MCT-M* then selects a set of winners to minimize the social cost through a reverse auction such that each cooperative task can be completed by a group of users, who belong to the same *compatible user group*.

In the multi-bid model, each task submitted by users is with a bid price, thus we can select winning task-bid pairs for each task independently. For any task $t_j \in T$, we process each *compatible user group* $\mathcal{G}_k, k = 1,2,\ldots,d$. In each iteration, we check if there are r_j users, who bid for t_j in $\mathcal{G}_k$. If so, we select r_j users from $\mathcal{G}_k$ with minimum total bid price, and the set is denoted as S_k. Then *MCT-M* selects the set with minimum $cost = \sum_{k \in S_i} b_k^j$ as the winner set for t_j from all $S_i, i = 1,2,\ldots,d$. We apply *VCG* based payment rule to determine the payment for each winning task-bid pair. A winning task-bid pair will be paid an amount equal to the benefit it introduces to the system, i.e., the difference between other users' minimum social cost with and without it:

$$p_i^j = cost(\bigcup_{i \in U_{\mathcal{G}}} \beta_i \setminus \{\beta_i^j\}) - (cost(\bigcup_{i \in U_{\mathcal{G}}} \beta_i) - b_i^j), \forall \beta_i^j \in \beta_S$$

Here function $cost()$ means the minimum social cost computed by *MCT-M*. Finally, we determine the payment for each winner i as $p_i = \sum_{\beta_i^j \in \beta_i \cap \beta_S} p_i^j$. The whole process is illustrated in Algorithm 4.

C. Mechanism Analysis

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

Lemma 5. *MCT-M is computationally efficient.*

Proof: It suffices to prove that both Algorithm 3 and Algorithm 4 are computationally efficient.

In Algorithm 3, the running time of m-*RP* (Line1-29) is dominated by sorting the users in U_i (Line12 or Line19). For each of m subset, m-*RP* performs the sorting. The worst case happens when all users are assigned to the same subset. In this case, m-RP takes $O(nlogn)$ time. Grouping the users in $S_{\mathcal{k}}$ (Line30) takes $O(\mathcal{k}^2)$ time. Thus Algorithm 3 takes $O(max\{nlogn, \mathcal{k}^2\})$ time.

In Algorithm 4, the running time of winner selection phase is dominated by sorting the users based on bid price in each *compatible user group* (Line8-11). For each task in T, the winner selection phase executes the sorting for each of d *compatible user group*. The worst case happens when all users are in the same *compatible user group*. In this case, the winner selection phase takes $O(m\mathcal{k}log\mathcal{k})$ time. In the payment determination phase, a process similar to winner selection phase is executed for each winning task-bid pair. Since there are at most $\sum_{j=1}^{m} r_j$ winning task-bid pairs, running time of the Algorithm 4 is bounded by $O((\sum_{j=1}^{m} r_j) m\mathcal{k}log\mathcal{k})$. ■

Lemma 6. *The reverse auction is optimal and individually rational.*

Proof: It is easy to know that the *reverse auction* can output the optimal solution of *SOCUS* problem in the multi-bid model. Since we adopt *VCG* payment rule, which is known as an individually rational auction, the *reverse auction* is individually rational. ■

Before analyzing the truthfulness of *MCT-M*, we first introduce the Theorem about m-*RP*.

Theorem 5. *([11, Theorem4.1])* For every value of m, *m-RP is truthful.*

The truthfulness in Theorem 5 means that no user can improve the chance of being selected into $S_{\mathcal{k}}$ by submitting a false *compatible user set*, no matter what others submit.

Lemma 7. *MCT-M is truthful.*

Proof: The compatibility-truthfulness can be guaranteed by Theorem 5. Since we adopt *VCG* payment rule, which is known as a cost-truthful auction, *MCT-M* is cost-truthful. ■

The above three lemmas together prove the following theorem.

Theorem 6. *MCT-M is computationally efficient, individually rational, and truthful, and the reverse auction is an optimal algorithm of SOCUS problem in the multi-bid model.*

V. Incentive Mechanism For The Single-bid Model

In this section, we consider the case where each user can submit a single bid price for all submitted tasks, and present an incentive mechanism for Multiple Cooperative Tasks in the Single-bid model (*MCT-S*) with considering the compatibility among users.

A. Mechanism design

Similar with *MCT-M*, *MCT-S* is a two-step mechanism. The grouping method is as same as that in *MCT-M*. Thus we focus on solving the *SOCUS* problem in the single-bid model in this subsection. Unfortunately, the following theorem shows that it is NP-hard to find the optimal solution.

Theorem 7. *The SOCUS problem in the single-bid model is NP-hard.*

Proof: As Theorem 2 shows, the *SOCUS* problem in the special case of single-bid model is equivalent to the *WSMC* problem. We can see that the *SOCUS* problem in the single-bid model is a generalization of the *WSMC* problem when each t_j only can be covered by the users who are within the same *compatible user group*. Since the *WSMC* problem is NP-hard, the *SOCUS* problem in the single-bid model is NP-hard. ■

Since the *SOCUS* problem in the single-bid model is NP-hard, we turn our attention to develop a polynomial algorithm. The main idea of *MCT-S* is selecting winners iteratively with minimum marginal cost for each task. Illustrated in Algorithm 5, the *reverse auction* still consists of the winner selection phase and the payment determination phase.

In the winner selection phase, *MCT-S* processes tasks in arbitrary fixed order. For each task t_j, we process each *compatible user group* $\mathcal{G}_k, k = 1,2,\dots,d$, iteratively. In each iteration, let S_k be the set of winners in $\mathcal{G}_k$ in current state. Let $Q_k \subseteq S_k$ be the set of winners, who bid for t_j. Then *MCT-S* checks if there are r_j users, who bid for t_j in $\mathcal{G}_k$. If so, we select additional $r_j - |Q_k|$users in $\mathcal{G}_k$ as winners, denoted by S'_k, with minimum marginal cost. The minimum marginal cost of $\mathcal{G}_k$ for t_j is denoted as $cost_{\mathcal{G}_k}^{t_j} = min \sum_{i \in S'_k} b_i$. For task t_j, *MCT-S* selects S'_k as the additional winner set with minimum $cost_{\mathcal{G}_k}^{t_j}$ among all $k = 1,2,\dots,d$. The winner selection phase terminates when all tasks have been processed.

Algorithm 5: Reverse Auction for Single-bid Model

Input: task set T, bid profile $\boldsymbol{B}$, compatible user group set $\mathcal{G}$, the set of $\mathscr{k}$ users $U_{\mathcal{G}}$

// Winner Selection Phase
$S \leftarrow \emptyset$; $\mathcal{G} \leftarrow \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_d\}$;
for $k \leftarrow 1$ to d **do**
  $S_k \leftarrow \emptyset$; $S_k \leftarrow \emptyset$;
  foreach $t_j \in T$ **do** $cost_{\mathcal{G}_k}^{t_j} \leftarrow 0$;
end
foreach $t_j \in T$ in arbitrary fixed order **do**
  foreach $k \leftarrow 1$ **to** d **do**
    $S_k \leftarrow \emptyset$;
    $Q_k \leftarrow \{i | i \in S_k, t_j \in \beta_i\}$;
    $num_k \leftarrow |\{i | i \in \mathcal{G}_k, t_j \in \beta_i\}|$;
    if $num_k \geq r_j$ **then**
      if $r_j \leq |Q_k|$ **then**
        break;
      else
        do
          $i \leftarrow \arg min_{i \in \mathcal{G}_k \setminus (Q_k \cup S_k), t_j \in \beta_i} b_i$;
          $S_k \leftarrow S_k \cup \{i\}$;
          $cost_{\mathcal{G}_k}^{t_j} \leftarrow cost_{\mathcal{G}_k}^{t_j} + b_i$;
        until $|Q_k| + |S_k| \geq r_j$;
      end
    else
      $cost_{\mathcal{G}_k}^{t_j} \leftarrow$;
    end
  end
  $k \leftarrow \arg min_{k \in \{1,2,\dots,d\}} cost_{\mathcal{G}_k}^{t_j}$;
  $S_k \leftarrow S_k \cup S_k$;
  $S \leftarrow S \cup S_k$;
end
//Payment Determination Phase
foreach $i \in U$ **do** $p_i \leftarrow 0$;
foreach $i \in S$ **do**
  foreach $t_j \in T$ in the same fixed order **do**
    Select winners from $U_{\mathcal{G}} \setminus \{i\}$ for t_j;
    Let $cost(U_{\mathcal{G}} \setminus \{i\})^{t_j}$ be the marginal cost for performing t_j without i;
    Let $cost(U_{\mathcal{G}})^{t_j}$ be the marginal cost for performing t_j with i;
    if $cost(U_{\mathcal{G}})^{t_j} < cost(U_{\mathcal{G}} \setminus \{i\})^{t_j}$ **then**
      $p_i \leftarrow max\{p_i, cost(U_{\mathcal{G}} \setminus \{i\})^{t_j} - (cost(U_{\mathcal{G}})^{t_j} - b_i)\}$;
    end
  end
end
return (S, $\mathbf{p}$);

In the payment determination phase, for each winner $i \in S$, *MCT-S* calls the winner selection phase to select winners from $U_{\mathcal{G}} \setminus \{i\}$ for all tasks iteratively. Let $cost(U_{\mathcal{G}} \setminus \{i\})^{t_j}$ be the marginal cost for performing t_j without *i*. Let $cost(U_{\mathcal{G}})^{t_j}$ be the marginal cost for performing t_j with *i*. If *i* is a winner for t_j, i.e., $cost(U_{\mathcal{G}})^{t_j} < cost(U_{\mathcal{G}} \setminus \{i\})^{t_j}$, we compute the maximum price of *i* to make the group including *i* can be selected instead of another group without *i*. We will prove that this price is a critical payment for user *i* later.

B. A Walk-Through Example

We use the example in Fig.4 to illustrate how the *reverse auction* of *MCT-S* works.

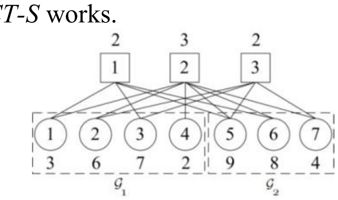


Fig. 4 An example illustrating how the *reverse auction* of *MCT-S* works, where the disks represent users, the squares represent tasks. The numbers below the disks represent the costs. The numbers above squares represent *cooperative index*. $\beta_1 = \{1,2\}$, $\beta_2 = \{2,3\}$, $\beta_3 = \{1,3\}$, $\beta_4 = \{2\}$, $\beta_5 = \{1,2,3\}$, $\beta_6 = \{1,2\}$, $\beta_7 = \{2,3\}$. The dotted squares represent *compatible user groups*. $\mathcal{G}_1 = \{1,2,3,4\}$, $\mathcal{G}_2 = \{5,6,7\}$.

Winner Selection:

- For task 1, $S = \emptyset$, $S'_1 = \{1,3\}, cost_1^1 = b_1 + b_3 = 10$, $S'_2 = \{5,6\}$, $cost_2^1 = b_5 + b_6 = 17$.
- For task 2, $S = \{1,3\}$, $S'_1 = \{2,4\}$, $cost_1^2 = b_2 + b_4 = 8$, $S'_2 = \{5,6,7\}$, $cost_2^2 = b_5 + b_6 + b_7 = 21$.
- For task 3, $S = \{1,2,3,4\}$, $S'_1 = \emptyset, cost_1^3 = 0$, $S'_2 = \{5,7\}$, $cost_2^3 = b_5 + b_7 = 13$.

During the payment determination phase, we directly give the winners when user *i* is excluded from the consideration, due to the space limitations.

Payment Determination:

- p_1: For task 1, winners are $\{5,6\}$, $p_1 = cost(\{5,6\})^1 - (cost(\{1,3\})^1 - b_1) = 10$. For task 2, additional winners are $\{7\}$, $cost(\{1,2,4\})^2 > cost(\{7\})^2$. For task 3, additional winners are $\emptyset$. Thus $p_1 = 10$.
- p_2: For task 1, winners are $\{1,3\}$, $cost(\{1,3\})^1 = cost(\{1,3\})^1$. For task 2, additional winners are $\{5,6,7\}$, $p_2 = cost(\{5,6,7\})^2 - (cost(\{2,4\})^2 - b_2) = 19$. For task 3, additional winners are $\emptyset$. Thus $p_2 = 19$.
- p_3: For task 1, winners are $\{5,6\}$, $p_3 = cost(\{5,6\})^1 - (cost(\{1,3\})^1 - b_3) = 14$. For task 2, additional winners are

$\{7\}$, $cost(\{1,2,4\})^2 > cost(\{7\})^2$. For task 3, additional winners are $\emptyset$. Thus $p_3 = 14$.

- p_4 : For task 1, winners are $\{1,3\}$, $cost(\{1,3\})^1 = cost(\{1,3\})^1$. For task 2, additional winners are $\{5,6,7\}$, $p_4 = cost(\{5,6,7\})^2 - (cost(\{2,4\})^2 - b_4) = 15$. For task 3, additional winners are $\emptyset$. Thus $p_4 = 15$.

C. Mechanism Analysis

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

Lemma 8. *MCT-S is computationally efficient.*

Proof: Since *MCT-S* adopts the same *compatible user grouping* method (Algorithm 3) of *MCT-M*, the first step of *MCT-S* takes $O(max\{nlogn, k^2\})$ time. In *reverse auction* step (Algorithm 5), the winner selection phase in the worst case takes $O(mklogk)$ time, which is as same as that in *MCT-M*. In the payment determination phase, a process similar to winner selection phase is executed for each winner. Since there are at most $min\{\sum_{j=1}^{m} r_j, k\}$ winners, the running time of Algorithm 5 is $O(min\{\sum_{j=1}^{m} r_j, k\}mklogk)$. ■

Lemma 9. *MCT-S is individually rational.*

Proof: We assume that user i is selected for task j in winner selection phase. Since the payment determination phase processes all tasks and *compatible user groups* in the same order, the output of winner selection before task j would not be changed. This means that, in the payment determination phase, for task j, it will obtain less or equal marginal cost to choose a group of additional winners including i than another group of additional winners without i, i.e., $cost(U_G)^{t_j} < cost(U_G\backslash\{i\})^{t_j}$. Hence we have $b_i < cost(U_G\backslash\{i\})^{t_j} - (cost(U_G)^{t_j} - b_i) \le p_i$. This is sufficient to guarantee $b_i < max_{t_j \in T}(cost(U_G\backslash\{i\})^{t_j} - (cost(U_G)^{t_j} - b_i)) = p_i$. ■

Lemma 10. *MCT-S is truthful.*

Proof: The compatibility-truthfulness can be guaranteed by Theorem 5. Based on Theorem 3, it suffices to prove that the selection rule of *MCT-S* is monotone and the payment p_i for each i is the critical value. The monotonicity of the selection rule is obvious as bidding a smaller value cannot push user i backwards in the sorting.

We next show that p_i is the critical value for i in the sense that bidding higher p_i could prevent i from winning the auction. Note that in the iteration of t_j, $p_i = cost(U_G\backslash\{i\})^{t_j} - (cost(U_G)^{t_j} - b_i)$. If user i bids $b_i > p_i$, the group of additional winners including i would be replaced by another group without i since $b_i > cost(U_G\backslash\{i\})^{t_j} - (cost(U_G)^{t_j} - b_i)$ implies $cost(U_G)^{t_j} > cost(U_G\backslash\{i\})^{t_j}$. Hence, user i would not win the auction for t_j. Based on line 36 in Algorithm 3, $p_i = max_{t_j \in T}(cost(U_G\backslash\{i\})^{t_j} - (cost(U_G)^{t_j} - b_i))$. User i would not win the auction because each $t_j \in T$ has chosen a group of additional winners without i. ■

The above three lemmas together prove the following theorem.

Theorem 8. *MCT-S is computationally efficient, individually rational and truthful in the single-bid model.*

VI. Learning User Compatibility

In Section IV-A, we have presented a *compatible user grouping* method based on m-RP. However, this *grouping* method may not effective in some situations. For example, the users may don't know the *compatible user set* exactly. In this case, an effective method is to learn the preferences of users based on the historical multiple cooperative mobile crowd sensing tasks. Moreover, m-RP will take long time to group users. We can utilize the neural network model to train the similarity of users in offline manner. From the perspective of performance, in order to achieve the truthfulness, m-*RP* excludes some users using random method before grouping, which may increase the social cost and overpayment ratio.

In this section, we present *User2Vec*, which fallows *Word2Vec* model [26] to mine the similarity between users, and group the users using *Density-Based Spatial Clustering of Applications with Noise* (*DBSCAN*) [29] according to this similarity.

A. User2Vec

We map each user to a high-dimensional vector, and use user compatible set as a training set to train the user vector by building *User2Vec* model.

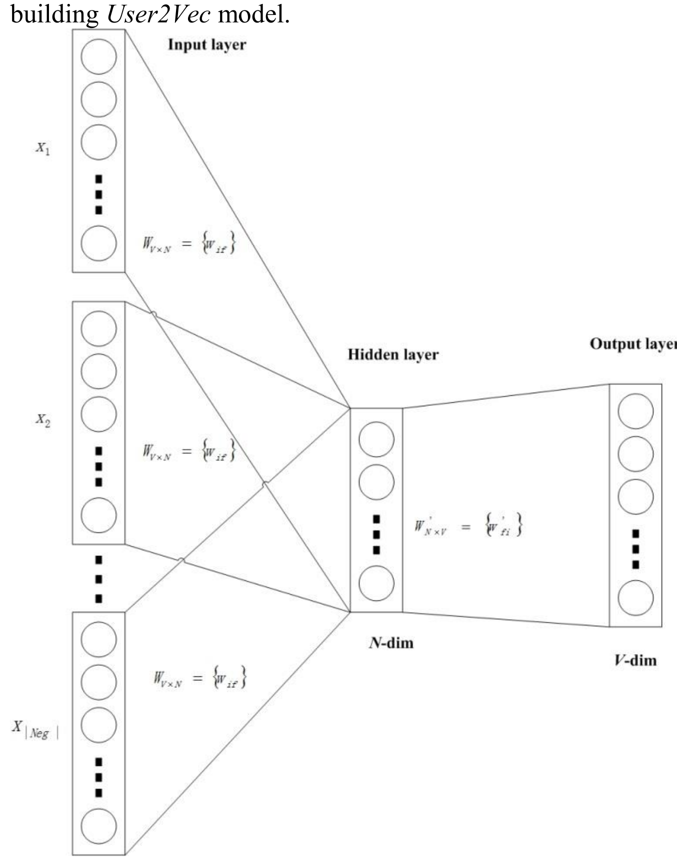


Fig. 5 *User2Vec* model

Fig.5 shows the *User2Vec* model with a multi-user vector set setting. In our setting, the user size is $V = |U|$. The size of hidden layer is N. The units on the adjacent layers are fully connected. The input is the multiple one-hot encoded vectors, where each vector represents a user. For each given user $i \in U$ only one out of V units $\{x_{i1}, \dots, x_{iV}\}$ will be 1, and all other units are 0. The weights between each input user vector and the output layer can be represented by a $V \times N$ matrix W. Each row of W is the N-dimension vector representation v_i of associated user of the input layer. Formally, row i of $\boldsymbol{W}$ is $v_i^T = \{w_{i1}, \ldots w_{if}, \ldots, w_{iN}\}$. For a set of users U' who work together to complete the task, *User2Vec* takes the users in U' and users in set $Neg= U'\backslash\{i\}, i \in U'$ as the target and the input one-hot vectors, respectively. The Hidden layer vector h is:

$$h = \frac{1}{|Neg|} W^T(x_1 + x_2 + \cdots + x_{|Neg|}) \quad (3)$$

$$= \frac{1}{|Neg|}(v_1 + v_2 + \cdots + v_{|Neg|})^T \quad (4)$$

From the hidden layer to the output layer, there is an $N \times V$ matrix $W' = \{w'_{fi}\}$. We compute a score s_i for each user $x_i \in U$:

$$s_i = v_i'^{\ T} h \quad (5)$$

where v'_i is the i-th column of the matrix W'. Then we use *softmax* (a log-linear classification model) [30] to obtain the posterior distribution of users, which is a multinomial distribution.

$$Pr(i|1,2,\ldots,V) = y_g = \frac{\exp(s_i)}{\sum_{i'=1}^{V}\exp(s_{i'})} \quad (6)$$

where y_g is the output of the i-th unit in the output layer. Substituting (4) and (5) into (6), we obtain

$$Pr(i|1,2,\ldots,V) = \frac{\exp(v_i'^{\ T} h)}{\sum_{i'=1}^{V}\exp(v_{i'}'^{\ T} h)} \quad (7)$$

Note that v_i and v'_i are two represents of the user i. v_i comes from rows of W, which is the input→hidden weight matrix, and v'_i comes from columns of W', which is the hidden→output matrix.

We derive the weight update equation for this model. The training objective (for one training sample) is to maximize (7), the conditional probability y_{i^*} of observing the actual output user o (denote its index in the output layer as i^*) given the input users with regard to the weights.

$$max\, Pr(o|1,2,\ldots,V) = max\, y_{i^*}$$
$$\Leftrightarrow max \log y_{i^*}$$
$$= s_{i^*} - \log \sum_{i'=1}^{V}\exp(s_{i'}) \quad (8)$$

Let $E = -log\, Pr(o|1,2,\ldots,V)$ be loss function, and the target is to minimize E. i^* is the index of the actual output user in the output layer.

We derive the update equation of the weights between hidden and output layers. Through taking the derivation of E with regard to i-th unit's net input s_i, we obtain

$$e_i = \frac{\partial E}{\partial s_i} = y_i - g_i \quad (9)$$

where $g_i = 1$ if $i = i^*$, otherwise $g_i = 0$. Note that this derivation is simply the prediction error e_i of the output layer.

Next we take the derivation on w'_{fi} to obtain the gradient on the hidden→output weights:

$$\frac{\partial E}{\partial w'_{fi}} = \frac{\partial E}{\partial s_i} \cdot \frac{\partial s_i}{\partial w'_{fi}} = e_i \cdot h_f \quad (10)$$

Therefore, using stochastic gradient descent, we obtain the weight updating equation for hidden→output weights:

$$w_{fi}'^{\ (new)} = w_{fi}'^{\ (old)} - \eta \cdot e_i \cdot h_f \quad (11)$$

or

$$v_i'^{\ (new)} = v_i'^{\ (old)} - \eta \cdot e_i \cdot h \quad \text{for } i=1,2,\ldots,V \quad (12)$$

where $\eta > 0$ is the learning rate, h_f is the f-th unit in the hidden layer. This update equation implies that we have to go through every possible user in U, check its output probability y_i, and compare y_i with its expected output g_i. If $y_i > g_i$, then we subtract a proportion of the hidden vector h (i.e., $v_1 + v_2 + \cdots + v_{|Neg|}$) from v'_i, thus making v'_i farther away from h. If $y_i < g_i$, which is true only if $g_i = 1$(i.e., $i = o$), we add h to v'_o, thus making v'_o closer to h. If y_i is very close to g_i, then according to the update equation, very little change will be made to the weights.

We then can pay attention to W. We take the derivation of E on the output of the hidden layer, obtaining

$$EH = \frac{\partial E}{\partial h_i} = \sum_{i=1}^{V} \frac{\partial E}{\partial s_i} \cdot \frac{\partial s_i}{\partial h_i} = \sum_{i=1}^{V} e_i \cdot w'_{fi} \quad (13)$$

where EH, an N-dim vector, is the sum of the user vectors in Neg, weighted by their prediction error.

Then we apply the following equation for every user i in Neg to update equation for input→hidden weights:

$$v_i^{(new)} = v_i^{(old)} - \frac{1}{|Neg|} \cdot \eta \cdot EH^T, \quad \text{for } i=1,2,\ldots,|Neg| \quad (14)$$

In the output layer, if the probability of a user i being output user is overestimated ($y_i > g_i$), then the input vector of the user set will tend to move farther away from the output vector of i. Conversely, if the probability of i being the output user is underestimated ($y_i < g_i$), then the input vector will tend to move closer to the output vector of i. If the probability of i is fairly accurately predicted, then it will have little effect on the movement of the input vector. The movement of the input vector is determined by the predication error of all vectors in U, the larger the predication error, the more significant effects a user will exert on the movement on the input vector of the user set.

In order to deal with the difficulty of having too many output vectors that need to be updated per iteration, we only update a sample of them.

Apparently, the output user (i.e., the ground truth, or positive sample) should be kept in our sample and gets updated, and we need to sample a few users as negative samples (negative sampling). A probabilistic distribution is needed for the sampling process, and it can be arbitrarily chosen. We denote this distribution as $P_n(u)$. The method in [27] can determine a good distribution empirically.

Instead of using a form of negative sampling that produces a well-defined posterior multinomial distribution, the following simplified training objective is capable of producing high-quality user embeddings [28]:

$$E = -log\sigma(v_o'^{\ T} h) - \sum_{i \in U_{neg}} log\sigma(-v_i'^{\ T} h) \quad (15)$$

where $U_{neg} = \{i|i = 1,\ldots,K\}$ is the set of users that are sampled based on $P_n(u)$, i.e., negative samples.

To obtain the update equations of the user vectors under negative sampling, we first take the derivative of E with regard to the net input of the output unit i:

$$\frac{\partial E}{\partial v_i'^{\ T} h} = \begin{cases} \sigma(v_i'^{\ T} h) - 1 & if\ \ u_i = u_o \\ \sigma(v_i'^{\ T} h) & if\ \ u_i \epsilon U_{neg} \end{cases}$$
$$= \sigma(v_i'^{\ T} h) - g_i \quad (16)$$

where g_i is the label of user i, $g_i = 1$ when i is a positive sample, $g_i = 0$ otherwise. Next we take the derivation of E with regard to the output vector of user i,

$$\frac{\partial E}{\partial v_i'} = \frac{\partial E}{\partial v_i'^{\,T} h} \cdot \frac{\partial v_i'^{\,T} h}{\partial v_i'} = (\sigma(-v_i'^{\,T} h) - g_i) h \quad (17)$$

which results in the following update equation for its output vector,

$$v_i'^{\,(new)} = v_i'^{\,(old)} - \eta \cdot (\sigma(-v_i'^{\,T} h) - g_i) \cdot h \quad (18)$$

which only needs to be applied to $i \epsilon \{o\} \cup U_{neg}$ instead of every user in U.

To backpropagate the error to the hidden layer and thus update the input vectors of users, we take the derivation of E with regard to the hidden layer's output, obtaining:

$$\begin{aligned} EH &= \frac{\partial E}{\partial h} \\ &= \sum_{i \in U_{neg} \cup \{o\}} \frac{\partial E}{\partial v_i'^{\,T} h} \cdot \frac{\partial v_i'^{\,T} h}{\partial h} \\ &= \sum_{i \in U_{neg} \cup \{o\}} (\sigma(-v_i'^{\,T} h) - g_i)\, \partial v_i' \quad (19) \end{aligned}$$

By plugging EH into (14) we obtain the update equation for the input vectors.

We get the final vector for each user based on the trained *User2Vec* model. A continuous space model works in terms of user vectors, where similar users are likely with similar vectors.

B. Grouping based on DBSCAN

In this subsection, we group the users based on the vectors processed through *User2Vec*. Since the number of *compatible user groups* is unknown, we use *DBSCAN* to group users. Given the user set U, *DBSCAN* is a density-based clustering algorithm which formulates a local density denoted as $density(i) = |N_{Eps}(i)|$ in the neighborhood of the *i-th* user. $N_{Eps}(i) = \{i' | \forall i' \in U \backslash \{i\}, distance(i', i) < Eps\}$ is the set of neighboring users in the neighborhood of within radius *Eps*, where $distance(i', i)$ is the Euclidean distance between i' and i in high dimensional vector space.

Given the minimal number of users in the neighborhood, *MinPts*, any user i is defined as a core user if $density(i) \geq MinPts$. Any user i is a noise user if $density(i) < MinPts$. For our settings, *MinPts* is determined by the *cooperative index* of tasks.

BDSCAN based Grouping works as follows:

(1) Choose arbitrary unvisited user, and find $N_{Eps}(i)$.

(2) If $density(i) \geq MinPts$, user i and $N_{Eps}(i)$ generate a new group together. Recursively process all unvisited users in the current group in the same way to expand the group.

(3) If $density(i) < MinPts$, mark user i as noise user.

(4) For other unvisited users, repeat step (1) to step (3) until all users are belong to a group or are marked as noise users.

VII. Performance Evaluation

We have conducted thorough simulations to investigate the performance of *MCT-M* and *MCT-S* for all three compatibility models. Due to the space limitations, we only give the numerical results under weak compatibility model. To investigate the performance of social optimization for *SOCUS* problem, we also implement the benchmark algorithms without considering the compatibility among users: *Benchmark-M* for multi-bid model and *Benchmark-S* for single-bid model, respectively.

We measure the number of winners, social cost, running time and overpayment ratio (a metric to measure the frugality of a mechanism [23], calculated by $\frac{\sum_{i \in S} p_i - Cost(S)}{Cost(S)}$), and reveal the impacts of the key parameters, including the number of users (n), the number of cooperative tasks (m) and *cooperative index* (r).

A. Simulation Setup

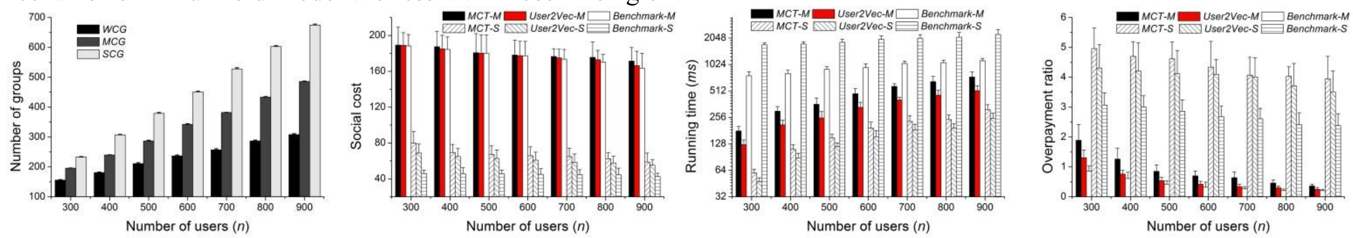
The simulations are based on Wikipedia vote network [21], which contains all the Wikipedia voting data for adminship election from the inception of Wikipedia till January 2008. Nodes in the network represent Wikipedia users and a directed edge from node i to node j represents that user i voted on user j. There are 7115 nodes and 103689 edges in the network. For our simulations, we select a set of users uniformly from whole Wikipedia vote network, and construct a sub-network only consisting of selected users and the edges among them. We set the *compatible user set* of arbitrary user as the set of users it voted on within the sub-network.

We set the default value of parameters as follows: The cost of each bid is uniformly distributed in [5, 10]. The *cooperative index* and the number of bidding tasks of each user are uniformly distributed in [2, 5] and [3, 5], respectively. The window of *User2Vec* is 5. Let n=300, m=10, $\hbar$=250, *Eps*=3, *MinPts*=5. However, we will vary the value of key parameters to explore the impacts of these parameters respectively. For convenience, we use *User2Vec-M* and *User2Vec-S* to represent the incentive mechanisms by learning user compatibility for multi-bid model and single-bid model, respectively. All the simulations were run on an Ubuntu 14.04.4 LTS machine with Intel Xeon CPU E5-2420 and 16 GB memory. Each measurement is averaged over 100 instances.

B. Impact of n

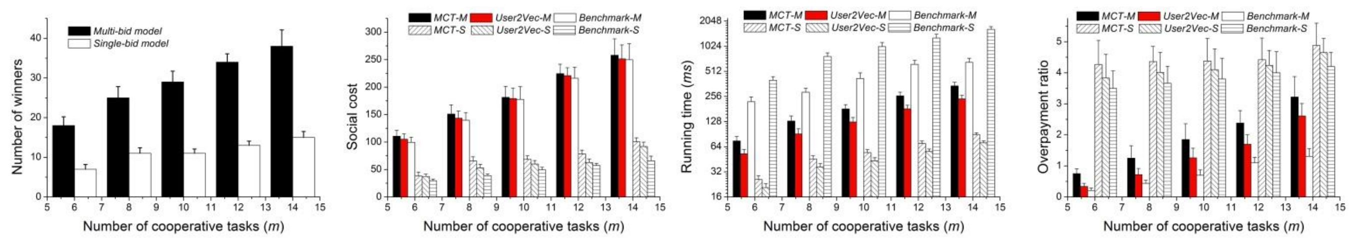
To investigate the scalability of designed mechanisms, we vary the number of users from 300 to 900, and select 80% users for each instance through m-RP for MCT, i.e., $\hbar = 0.8 * n$. As shown in Fig.6, the number of *compatible user groups* goes up under all three compatibility models when the number of users increases. There are 2.5, 1.75 and 1.32 users in each *WCG*, *MCG* and *SCG* on average, respectively. The social cost decreases with increasing user number since the platform can find more cheap users. However, the change of social cost is very slight because in our system model, the user number needs to be large enough in order to complete all cooperative tasks. The social cost of *MCT-M* and *User2Vec* are very close to that of *Benchmark-M* (only 1.8% more social cost than *Benchmark-M* on average) since the reverse auction for multi-bid model (Algorithm 4) can output optimal solution. However, *MCT-S* outputs 48.9% more social cost than *Benchmark-S* on average. Note that *User2Vec-S* can reduce 8.24% of social cost comparing with *MCT-S* since all users can pass to the *reverse auction* step in *User2Vec-S*. Moreover, the designed mechanisms are computational efficient since the running time of *MCT-M* and *MCT-S* is bounded by 0.8s and

0.4s, respectively, even there are 900 users. The running time of User2Vec-M and User2Vec-S are only 81.8% and 72.1% of MCT-M and MCT-S, respectively, since the training of user vector similarity can be executed offline. Based on the frugality theory, the overpayment ratio depends on the competition among users. As seen from Fig.6(d), the overpayment ratio of all incentive mechanisms decrease because the competition among users intensify when there are more users. The overpayment ratio of the incentive mechanisms in multi-bid model are less than those in single-bid model. Obviously, the competition of users in multi-bid model is more than that in single-bid model since the incentive mechanisms in multi-bid model can select winning task-bid pairs independently from all task-bid pairs and the cost of each task follows the identical distribution. Note that the *User2Vec* based methods can output less overpayment ratio because there are more users in the *reverse auction* step comparing with m-RP based methods.



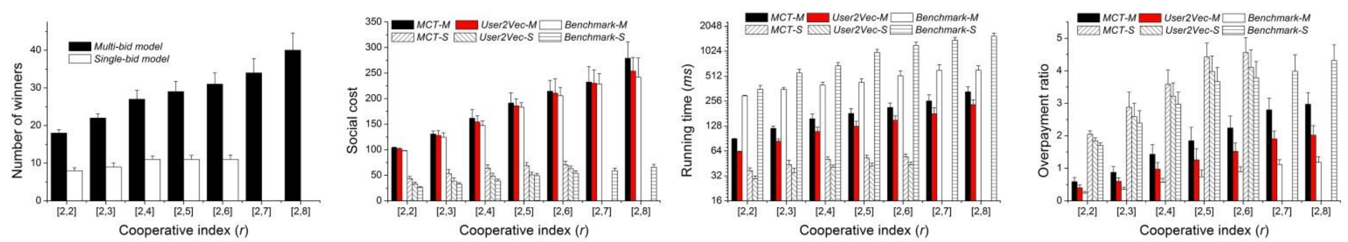
(a) (b) (c) (d)

Fig. 6 Impact of the number of users (n): (a) Number of groups. (b) Social cost. (c) Running time. (d) Overpayment ratio



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

Fig. 7 Impact of the number of cooperative tasks (m): (a) Number of winners. (b) Social cost. (c) Running time. (d) Overpayment ratio



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

Fig. 8 Impact of cooperative index(r): (a) Number of winners. (b) Social cost. (c) Running time. (d) Overpayment ratio

C. Impact of m

The number of cooperative tasks can depict the workload of mobile crowd sensing system. We fix $n = 300$, $k = 250$, and vary *m* from 6 to 14. As shown in Fig.7, the number of winners and the social cost increase severely in all incentive mechanisms with increasing *m* since the platform needs more users to complete the tasks. The winners in multi-bid model are much more than that of single-bid model because any user will be the winner if one of the task-bid pairs it submits is selected in the multi-bid model. Accordingly, the social cost of multi-bid model is more than that of single-bid model since the cost of each winner follows the identical distribution in our settings. The running time also increases with increasing tasks. However the running time of *MCT-M* and *MCT-S* are still lower than 0.4s and 0.1s when there are 300 users and 14 cooperative tasks, respectively. The overpayment ratio also increases since the platform needs to recruit more users to perform tasks, which mitigates the competition among users accordingly.

D. Impact of r

To investigate the performance for the tasks associated with different cooperative levels, we vary the distribution interval of *cooperative index* from [2, 2] to [2, 8]. As can be seen from Fig.8, *MCT-S* and *User2Vec-S* cannot output the solution when the *cooperative index* is too large (the upper limit of distribution interval exceeds 7). Both the winners and the social cost increase with increasing cooperative level since the platform needs more users to perform each cooperative task averagely. *MCT-M* and *MCT-S* output 6.7% and 52.6% more social cost than benchmark algorithms, respectively. The social cost can be reduced further by machine learning based grouping method. The running time and overpayment ratio also increase when the *cooperative index* goes up. The running time of *MCT-S* is only 32.9% of that of *MCT-M*, while the overpayment of *MCT-M* is much less than that of *MCT-S*. For both bid models, *User2vec* based incentive mechanisms can reduce the overpayment ration comparing to the *MCTs* since there are more users to compete the winners in auction.

VIII. Related Work

Many incentive mechanisms for mobile crowd sensing have been proposed thus far. Yang *et al*. proposed two different models for smartphone crowd sensing [9]: 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. Feng *et al*. [7] formulated the location-aware collaborative sensing problem as the *winning bids determination problem*, and presented a truthful auction using the proportional share allocation rule proposed in [15]. Koutsopoulos designed an optimal reverse auction [14], considering the data quality as *user participation level*. However, the *quality indicator*, which essentially measures the relevance or usefulness of information, is empirical and relies on users' historical information. Zhao *et al*. [16] investigated the online crowdsourcing scenario where the users submit their profiles to the crowdsourcer when they arrive. The objective is selecting a subset of users for maximizing the value of the crowdsourcer under the budget constraint. They designed two online mechanisms, *OMZ, OMG* for different user models. Zhang *et al*. proposed IMC [17], which consider the competition among the requesters in crowdsourcing. The incentive mechanisms considering the biased requesters were proposed in [25]. However, all above works focus on the multiple independent task scenarios, where each task only needs one user to perform.

Some works aim to the single cooperative task scenario, where the task requires a group of users to perform cooperatively. Xu *et al*. proposed truthful incentive mechanisms for the mobile crowd sensing system where the cooperative task is time window dependent, and the platform has strong requirement of data integrity [8]. Furthermore, they studied the budget feasible mechanisms for the same crowd sensing system [20]. Luo *et al*. designed the truthful mechanisms for multiple cooperative tasks [19, 24]. However, they don't consider the compatibility among users.

Word2Vec was usually used in natural language processing and recommendation systems. Mikolov *et al.* proposed two novel model architectures for computing continuous vector representations of words from very large data sets [26]. Ester *et al.* proposed a clustering algorithm, *DBSCAN*, which does not need to define the number of clusters [29]. However, no one use *Word2Vec* model and *DBSCAN* for user grouping in mobile crowd sensing.

Overall, there is no off-the-shelf incentive mechanism designed in the literature for the mobile crowd sensing system, where there are multiple cooperative tasks, and each of tasks requires a group of compatible users to perform.

IX. Conclusion and Future Work

In this paper, we have designed the incentive mechanisms for the mobile crowd sensing system with multiple cooperative tasks. We use real-life relationships from social networks to model the compatibility relationships. We have presented two bid models and three compatibility models for this new scenario, and designed two incentive mechanisms: *MCT-M* and *MCT-S* to solve the *SOCUS* problem for the two bid models, respectively. We have presented a user grouping method through neural network model and clustering algorithm for the situations, where the users don't know the *compatible user set* exactly. Through both rigorous theoretical analyses and extensive simulations, we have demonstrated that the proposed incentive mechanisms achieve computational efficiency, individual rationality and truthfulness. Moreover, *MCT-M* can output the optimal solution. By using neural network and clustering algorithm for user grouping, the proposed incentive mechanisms can reduce the social cost and overpayment ratio further with less grouping time.

In the future, we plan to construct different compatibility models according to different indicators, such as the geographical distance and user quality, for some specific mobile crowd sensing applications. In addition, we plan to design other machine learning based grouping methods, and valuate the accuracy of grouping by real-world mobile crowd sensing systems.

Acknowledgment

This work has been supported in part by the NSFC (No. 61472193, 61872193, 61502251), and NSF (No. 1444059, 1717315).

References

[1] IDC, Worldwide smartphone forecast update, 20162020: December 2016, http://www.idc.com/getdoc.jsp?containerId=US42060116, 2016.

[2] L. G. Jaimes, I. J. Vergara-Laurens, A. Raij, "A survey of incentive techniques for mobile crowd sensing," *IEEE Internet of Things Journal,* vol. 2, no. 5, pp. 370-380, Oct. 2015.

[3] Sensorly, Sensorly, http://www.sensorly.com (2014).

[4] X. Zhang, Z. Yang, C. Wu, W. Sun, Y. Liu, K. Liu, "Robust trajectory estimation for crowdsourcing-based mobile applications," *IEEE TPDS*, vol.25, no.7, pp.1876-1885, July. 2014.

[5] P. Zhou, Y. Zheng, M. Li, "How long to wait?: predicting bus arrival time with mobile phone based participatory sensing," *Proc. ACM MobiSys*, pp. 379-392, 2012.

[6] G. Cardone, L. Foschini, P. Bellavista, A. Corradi, C. Borcea, M. Talasila, R. Curtmola, "Fostering participaction in smart cities: a geo-social crowdsensing platform," *IEEE Communications Magazine,* vol. 51 no. 6, pp. 112-119, June. 2013.

[7] Z. Feng, Y. Zhu, Q. Zhang, L.M. Ni, A.V. Vasilakos, "TRAC: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing," *Proc. IEEE INFOCOM* , pp. 1231-1239, 2014.

[8] J. Xu, J. Xiang, and D. Yang, "Incentive mechanisms for time window dependent tasks in mobile crowdsensing," *IEEE Trans. Wireless Commun*, vol. 14, no. 11, pp. 6353–6364, Nov. 2015.

[9] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," *Proc. ACM MobiCom*, pp.173-184, 2012.

[10] L. Blumrosen and N. Nisan. "Combinatorial auctions," *Algorithmic Game Theory*, pp. 267-298, 2007.

[11] N. Alon, F. Fischer, A. Procaccia, et al. "Sum of us: Strategyproof selection from the selectors," *Proc. ACM Conference on Theoretical Aspects of Rationality and Knowledge*, pp. 101-110, 2012.

[12] S. J. Brams and P. C. Fishburn. "Approval Voting," *Springer*, 2nd edition, 2007.

[13] R. Myerson, “Optimal auction design,” *Math. of Operations Research*, vol. 6, no. 1, pp. 58-73, Feb. 1981.

[14] I.Koutsopoulos, "Optimal incentive-driven design of participatory sensing systems," *Proc. IEEE INFOCOM*, pp. 1402-1410, 2013.

[15] Y. Singer, “Budget feasible mechanisms,” *Proc. IEEE FOCS*, pp.765–774, 2012.

[16] D. Zhao, X. Li, and H. Ma, “How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint,” *Proc. IEEE INFOCOM*, pp. 1213-1221, 2014.

[17] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, “Truthful incentive mechanisms for crowdsourcing,” *Proc. IEEE INFOCOM* , pp.2830-2838, 2015.

[18] 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, Mar. 2017.

[19] S. Luo, Y. Sun, Z. Wen and Y. Ji, "C2: Truthful incentive mechanism for multiple cooperative tasks in mobile cloud," *Proc. IEEE ICC*, pp. 1-6, 2016.

[20] J. Xu, J. Xiang, X. Chen, et al. “ODMBP: behavior forwarding for multiple property destinations in mobile social networks,” *Mobile Information Systems*, 2016.

[21] J. Leskovec, D. Huttenlocher, J. Kleinberg. “Predicting positive and negative links in online social networks,” *Proc. ACM WWW*, pp.641-650, 2010.

[22] S. Rajagopalan, V. Vazirani. “Primal-dual RNC approximation algorithms for (multi)-set (multi)-cover and covering integer programs,” *Proc. 34th IEEE Annual Symposium on Foundations of Computer Science*, pp. 322-331, 1993.

[23] A. Karlin, and D. Kempe, “Beyond vcg: Frugality of truthful mechanisms,” *Proc. IEEE FOCS*, pp. 615-624, 2005.

[24] S. Luo, Y. Sun, Y. Ji, et al. “Stackelberg game based incentive mechanisms for multiple collaborative tasks in mobile crowdsourcing,” *Mobile Networks and Applications*, vol. 21, no. 3, pp. 506-522, Dec. 2016.

[25] J. Xu, H. Li, Y. Li, D. Yang, and T. Li, “Incentivizing the Biased Requesters: Truthful Task Assignment Mechanisms in Crowdsourcing”, *Proc. IEEE SECON*, pp. 1-9, 2017.

[26] T. Mikolov, K Chen, G Corrado, and J Dean. “Efficient estimation of word representations in vector space”, *Computer Science,* 2013.

[27] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality”, *Advances in Neural Information Processing Systems*, pp. 3111-3119, 2013.

[28] Y. Goldberg, and O. Levy, “word2vec explained: deriving mikolov et al.'s negative- sampling word-embedding method”, *arXiv: 1402.3722*, 2014.

[29] M. Ester, H.P. Kriegel, J.Scander, X. Xu, “A density-based algorithm for discocering clusters in large spatial databases with noise”, Proc. *The Second International Coference on Knowledge Discovery and Data Mining*, pp. 226-231, 1996.

[30] B. Gao, L. Pavel, “On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning[J]”, arXiv preprint arXiv:1704.00805, 2017.

[31] J. Xu, Z. Rao, L. Xu, D. Yang, and T. Li, “Mobile crowd sensing via online Communities: Incentive mechanism for multiple cooperative tasks,” Proc. *IEEE MASS,* pp. 171-179, 2017.

[32] Z. Wang, X. Pang, Y. Chen, H. Shao, Q. Wang, L. Wu, H. Chen, and H. Qi, “Privacy-preserving Crowd-sourced Statistical Data Publishing with An Untrusted Server”, *IEEE Transactions on Mobile Computing*, DOI: 10.1109/TMC.2018.2861765, 2018,

[33] Z. Wang, J. Hu, R. Lv, J. Wei, Q. Wang, D. Yang, and H. Qi, “Personalized Privacy-preserving Task Allocation for Mobile Crowdsensing”, *IEEE Transactions on Mobile Computing*, DOI: 10.1109/TMC.2018.2861393, 2018,