BundleSense: A Task-Bundling-Based Incentive Mechanism for Mobile Crowd Sensing

Yifan Zhang
School of Computer Science and Engineering
South China University of Technology
Guangzhou, China
yifanzhang.scut@foxmail.com

Xinglin Zhang
School of Computer Science and Engineering
South China University of Technology
Guangzhou, China
zhxlinse@gmail.com

***Abstract*—Mobile crowd sensing (MCS) has become a powerful sensing paradigm that allows requesters to outsource location-based sensing tasks to a crowd of participating users carrying smart mobile devices. Aware of the paramount importance of incentivizing participation for MCS systems, researchers have proposed a wide variety of incentive mechanisms. Most of these mechanisms assume that the MCS platform can collect sufficient budget to recruit users, and hence only focus on incentivizing users efficiently. In this work, we consider MCS systems where the budget of a single task is insufficient for recruiting a user. Commonly, a task requester with a simple task (e.g., inquiring a photo of a restaurant) is willing to provide a low budget, while a user would like to earn a higher reward for his effort in completing a task (e.g., traveling a long distance to take a photo). To address this disparity issue between requesters and users, we propose a novel task-bundling-based two-stage incentive mechanism to incentivize both requesters and users. Through rigorous theoretical analysis and extensive simulations, we demonstrate that the proposed incentive mechanism satisfies the properties of computational efficiency, individual rationality, budget balance, truthfulness, and constant competitiveness.**

***Index Terms*—Mobile crowd sensing, incentive mechanism, task bundling, two-stage auction.**

I. Introduction

With the proliferation of smart mobile devices possessing powerful sensing, networking, and computing capabilities, mobile crowd sensing (MCS) has become a prevalent sensing paradigm that can recruit a crowd of mobile users to perform a wide variety of location-based sensing tasks [1]. Examples of these tasks include urban traffic monitoring [2], object tracking [3], and environment monitoring [4]. With the recent advances of IoT technologies, MCS is expected to provide more novel spatial-temporal sensing applications for smart cities and cover almost every aspect of our lives.

Incentive mechanisms are of paramount importance for M-CS systems to attract sufficient participants. As users consume various resources, such as battery power and data transmission cost, and endure the risk of privacy leakage when undertaking sensing tasks, it is essential to provide them with sufficient rewards by delicate incentive design. Therefore, a great many incentive mechanisms have been proposed in the literature [5]–[10]. These works mostly assume that the budget provided by the platform or requester is sufficient to attract users to perform sensing tasks. Thus they focus on the characteristics of users when designing mechanisms given the budget.

However, in many MCS scenarios, the task of a requester is usually simple, e.g., a requester may need a photo of a nearby restaurant to check whether it is open or not, and then decides whether to eat at that restaurant after work. Naturally, the requester is unwilling to pay a lot for this photo (it is unacceptable if it costs more than his lunch). On the other hand, a mobile user may be far away from this restaurant, and needs to walk a long distance to complete the task. Hence, the user expects to receive a relatively high reward for completing such tasks. As a result, this disparity of expected payment and reward is very likely to cause that no user is willing to perform these small sensing tasks and the requesters cannot obtain the expected sensing data.

To address this essential issue, in this paper, we propose a new incentive framework where both requesters and users can be effectively motivated to participate in MCS. Inspired by group buying services on the radio spectrum sharing [11], [12], we assume that nearby tasks can be voluntarily bundled together to obtain and share services from users. In this way, the budgets of tasks in one bundle can be aggregated to form a total budget that is high enough to attract mobile users, although no individual task requester can afford the cost of recruiting a user. We consider that there is an agent in each bundle as a representative. The agent runs a mechanism to decide which tasks should be incorporated (these tasks are termed as winning tasks) and collect their budgets as the bundle's budget. A double auction is then conducted between agents and users. Winning users will be rewarded and responsible for performing the winning tasks in the corresponding bundle. At the same time, each winning agent will charge the winning tasks in his bundle according to the payment for the user.

Although group-buying-based mechanisms have been studied in spectrum auctions and a similar novel concept "task bundling scheme" has been proposed by some existing works

This work was supported in part by the National Natural Science Foundations of China under Grant 61632013 and 61872149, in part by the Natural Science Foundations of Guangdong Province for Distinguished Young Scholar under Grant 2018B030306010, in part by the Pearl River S&T Nova Program of Guangzhou under Grant 201806010088, and in part by the Guangdong Special Support Program under Grant 2017TQ04X482. (*Corresponding Author: Xinglin Zhang.*)

in MCS [13], [14], no existing mechanism can be directly applied to our scenario and the following characteristics make our work challenging.

First, a well-designed incentive mechanism needs to guarantee the crucial property of truthfulness, which prevents the auction participants from disrupting the market through manipulating bidding prices. Most existing works have fixed budgets for sensing tasks, and only consider the bidding behaviors of users when designing truthful mechanisms. However, in our considered scenarios, users, agents, and task requesters all participate in the auction and can bid freely. Hence it becomes more complicated to design a triple truthfulness mechanism, which ensures the truthfulness of all the three kinds of participants at the same time. Furthermore, it is also expected to design an incentive mechanism with a constant competitive ratio compared to the optimal mechanism with respect to the system optimization goal, which is more challenging in our scenario as we design a two-stage mechanism which consists of two auction processes.

Second, as one user performs multiple tasks which share the payment for recruiting the user, it is bound to cause time delay on completing these tasks (i.e., the waiting time for the user to complete all winning tasks in the corresponding bundle). The time delay may affect the task requester's satisfaction with the sensing service and thus diminish the task's utility. Therefore, we need to quantify the impact of time delay on the task's utility and take into account this factor when making decisions. It thus makes the incentive mechanism design more difficult as we need to determine a suitable number of winning tasks and the corresponding winning prices of winning tasks.

To accommodate the above challenges, we propose a two-stage incentive mechanism. The main idea is that, in Stage I, tasks with different location requirements form different bundles. To increase the budget for each bundle's agent, we design an algorithm to divide each bundle's tasks into a reference set and a candidate set, where the reference set determines the agent's budget and winners are chosen from the candidate set. In Stage II, agents and users submit their budgets and bids simultaneously. To optimize the platform's utility and ensure the truthfulness of both agents and users, we design an efficient mechanism to select agent-user pairs. Through theoretical analysis and simulations, we prove that the proposed mechanisms satisfy the desirable properties of computational efficiency, individual rationality, budget balance, truthfulness, and constant competitiveness.

The remainder of this paper is organized as follows: Section II presents the related work. In Section III, we describe the system model and formulate the problem as a two-stage auction. We then elaborate on the proposed mechanism in Section IV and present the evaluation results in Section V. Finally, we conclude this paper in Section VI.

II. Related Work

Researchers have spent much effort in designing incentive mechanisms for MCS [15]. Among the various types of mechanisms, reverse auction based mechanisms are most prevalent [5]–[10]. These mechanisms focus on the user's strategic behavior in the MCS system, and mostly assume that the budget for recruitment is sufficiently high. Our work differs from these works in that we observe the necessity of cooperation of task requesters with low budgets and thus propose a new incentive mechanism framework for MCS, which jointly considers the behaviors of requesters and users.

Double auction is applied in this study. Considering the theoretical performance, Deshmukh *et al.* [16] propose a technique to convert α-competitive basic auctions into 2α-competitive double auctions. This theory is widely used in designing double spectrum auction. Zhai *et al.* [17] propose a double auction based mechanism to improve the networks' benefit with high energy efficiency. Zhang *et al.* [18] propose a double auction mechanism to ensure fair trading for service. Compared with these works, our work addresses the incentive mechanism design problem with three parties in MCS. The proposed mechanism allows requesters, users, and agents to bid fairly and ensure that all participants are truthful.

Recently, the group buying scheme is considered when designing spectrum auctions in cognitive radio networks and incentive mechanisms in MCS. Lin *et al.* [11] propose a novel three-stage mechanism TASG based on group buying. Based on this, Huang *et al.* [19] observe that in MCS, there exists a mismatch between requesters' low budgets and workers' high prices. Thus they extend TASG for MCS and propose TGBA. In addition, a similar concept "task bundling" is also used in MCS. Wang *et al.* [14] propose a truthful incentive mechanism based on task bundling and tackle the problem of unbalance participation. Xie *et al.* [13] combine "task bundling scheme" and "rating system" to design an incentive mechanism. Although the aforementioned works use the "task bundling scheme" in different scenarios, these mechanisms cannot adapt to our scenario where we not only need the tasks to share payment and sensing capability of users, but also aim to guarantee all participants' truthfulness and the mechanism's competitiveness.

III. Preliminaries

In this section, we first describe the task bundling scenario in MCS and formulate the problem of recruiting mobile users as a two-stage auction. Then, we introduce the desirable properties for incentive mechanisms in MCS.

A. Problem Formulation

As shown in Fig. 1, a task bundling scenario consists of task requesters, agents, mobile users and a platform. Each requester submits a sensing task at a time. Tasks form different bundles thus they can use an accumulated budget to recruit users for improving the possibility of task completion. An agent acts as a representative for a bundle of tasks to directly negotiate with the platform for recruiting users. Mobile users act as workers who are interested in undertaking sensing tasks and they also negotiate with the platform for selling their sensing capability. The MCS platform, which may reside in the cloud, is responsible for matching agents and users, such that the

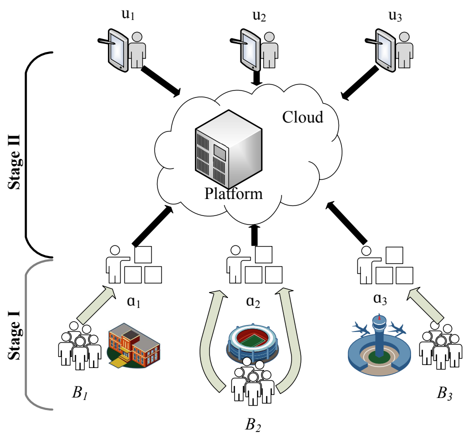


Fig. 1. A two-stage auction framework.

sensing tasks can be assigned to the suitable users and at the same time the platform achieves a high utility.

Mathematically, we consider the scenario where there are n bundles of tasks $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, ..., \mathcal{B}_n\}$ and there is an agent for each bundle. Thus we have n agents $\mathcal{A} = \{a_1, a_2, ..., a_n\}$. In the bundle $\mathcal{B}_j$, there are n_j tasks $\mathcal{B}_j = \{t_j^1, t_j^2, ..., t_j^{n_j}\}$. A task $t_j^k \in \mathcal{B}_j$ is associated with a tuple $(b_j^k, v_j^k, p_j^k, \mu_j^k)$, where the budget b_j^k is the task's maximum payment for the sensing data, the valuation v_j^k is the value of the sensing data that is only known to the task's requester, the payment p_j^k is the reward given to the user who completes the task, and the utility μ_j^k reflects the requester's utility for participating in MCS. There are a crowd of mobile users $\mathcal{U} = \{u_1, u_2, ..., u_m\}$ interested in performing sensing tasks. The user $u_i \in \mathcal{U}$ is identified by a tuple (v_i, s_i, r_i, μ_i), where the reserved price v_i reflects the true cost for performing sensing tasks and is only known to the user, the price s_i is the bidding price submitted to the platform, the reward r_i is the payment received from the platform for completing sensing tasks, and μ_i is the utility of the user for participating in MCS. We assume that a user serves for at most one bundle of tasks here, since each bundle is usually spatially clustered and thus is convenient for the user to complete.

We model the interaction between requesters, agents, and users as a two-stage auction. In Stage I, an auction is conducted for determining winning tasks and budget in each bundle, while in Stage II, a double auction is conducted between agents and users, where the agents use the budgets obtained in the first stage to recruit users.

Specifically, in Stage I, tasks with similar location requirements are collected into the same bundle and an agent is generated to represent the bundle. The requester of the task t_j^k submits a bid to the agent a_j. The agent a_j decides the set of winning tasks $\mathcal{W}_j$ and the total budget b_j of the bundle $\mathcal{B}_j$. These agents then participate in Stage II, which generates the final agent-user matching result $\Lambda = \{(a_j, u_i) | j \in [1, 2, ..., n], i \in [1, 2, ..., m]\}$. If the agent a_j is successfully matched with a user (i.e., $a_j \in \Lambda_a = \{a_j | (a_j, u_i) \in \Lambda\}$) by offering the payment p_j, the corresponding winning tasks in $\mathcal{W}_j$ will receive the sensing data and each task $t_j^k \in \mathcal{W}_j$ will be charged a payment p_j^k. Thus the utility of the agent a_j can be defined as:

$$\mu_j = \begin{cases} b_j - p_j, & a_j \in \Lambda_a, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The utility of the task is affected by the number of winning tasks in its bundle. If the number of winners is too large, it will take a long time for the user to complete the winning tasks in the bundle. Hence we assume that the task's utility is inversely proportional to the number of winning tasks in its bundle and define the time delay function $\lambda(\kappa) \in (0, 1]$:

$$\lambda(\kappa) = 1 - \frac{\kappa - 1}{2M}, \quad 1 \leq \kappa \leq M, \tag{2}$$

where κ is an integer less than the maximum bundle capacity M. The value of M depends on the task's timeliness requirement. For example, in [20], the photo-taking task needs sufficient space-time variety and hence M can be set to a large value. Conversely, if a task requires a quick response, then M should be set to a small value. Taking the time delay of tasks into account, the utility of a task $t_j^k \in \mathcal{B}_j$ is defined as:

$$\mu_j^k = \begin{cases} v_j^k \cdot \lambda(|\mathcal{W}_j|) - p_j^k, & t_j^k \in \mathcal{W}_j \text{ and } a_j \in \Lambda_a, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

In Stage II, agents act as buyers (with a set of budgets $\{b_1, b_2, ..., b_n\}$) and users act as sellers (with a set of bids $\{s_1, s_2, ..., s_m\}$). The platform matches users and agents strategically and finds a matching Λ. The user $u_i \in \Lambda_u = \{u_i | (a_j, u_i) \in \Lambda\}$ will be given a reward r_i which is not less than his bid. The utility is the difference of his reward and his reserved price. If $u_i \notin \Lambda_u$, his utility will be zero. Therefore, the utility of u_i can be defined as:

$$\mu_i = \begin{cases} r_i - v_i, & u_i \in \Lambda_u, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

For each matched pair $(a_j, u_i) \in \Lambda$, the agent a_j will recruit the user u_i to collect sensing data. The platform's utility is defined as the difference of agents' payments and users' rewards:

$$\mu = \sum_{a_j \in \Lambda_a} p_j - \sum_{u_i \in \Lambda_u} r_i. \tag{5}$$

For each $(a_j, u_i) \in \Lambda$, the value of p_j must be not smaller than the value of r_i, which ensures that the utility of the platform is nonnegative.

Table I lists frequently used notations.

B. Desirable Properties

In this work, our purpose is to design task-bundling-based two-stage incentive mechanisms, which are expected to satisfy the following properties.

- **Computational Efficiency:** An incentive mechanism is computationally efficient if it returns the result in polynomial time.

TABLE I
MAIN NOTATIONS

Symbol	Description
$\mathcal{U}, u_i, m$	set of users, ith user, and number of users
$\mathcal{B}, \mathcal{B}_j, n, n_j$	set of bundles, jth bundle, number of bundles, number of tasks in the jth bundle
t_j^k	kth task in jth bundle
$\mathcal{W}_j, w_j$	set of winners and number of winners in $\mathcal{B}_j$
$\mathcal{A}, a_j$	set of agents, jth agent w.r.t. jth bundle
$v_j^k, b_j^k, p_j^k, \mu_j^k$	task t_j^k's valuation, budget, payment and utility
v_i, s_i, r_i, μ_i	user u_i's reserved price, bid, reward and utility
b_j, p_j, μ_j	agent a_j's budget, payment and utility
Λ	matching result between users and agents
M	maximum bundle capacity

- **Individual Rationality:** An incentive mechanism is individually rational if the utility of each participant is nonnegative. The proposed mechanisms should make sure that users, agents, and task requesters are all rational.
- **Budget Balance:** An incentive mechanism is budget balanced if the utility of the auctioneer (i.e., the platform here) is nonnegative.
- **Truthfulness:** An incentive mechanism is truthful if a bidder cannot improve his utility by submitting a bidding price deviating from his true value, no matter what the others' bidding prices are. Here, the bidders include users, agents, and requesters, who are game-theoretic and tend to manipulate their bidding prices so as to maximize their received payments. Therefore, the mechanism needs to be triple truthful for users, agents, and task requesters.
- **Competitiveness:** Since the proposed mechanism is divided into two stages, we discuss the competitiveness of these two stages separately. In Stage I, the metric of the algorithm performance is the amount of budget the agent collects. A mechanism Ψ is said to be α-competitive if, for any bundle of tasks $\mathcal{B}$, these is a budget vector $\mathbf{b} = \{b_j^1, b_j^2, ..., b_j^{n_j}\}$, such that the expected budget obtained by Ψ satisfies $\mathbb{E}[\Psi(\mathbf{b})] \geq \Phi(\mathbf{b})/\alpha$, where Φ is the optimal mechanism. Similarly, in Stage II, the utility of the platform is the optimization objective of the mechanism. Given the budget vector $\mathbf{b}$ of all agents and the bid vector $\mathbf{s}$ of all users, we say that Ψ is α-competitive if the platform's expected utility obtained by the mechanism Ψ satisfies $\mathbb{E}[\Psi(\mathbf{b}, \mathbf{s})] \geq \Phi(\mathbf{b}, \mathbf{s})/\alpha$, where Φ is the optimal mechanism.

IV. INCENTIVE MECHANISM DESIGN

In this section, we demonstrate the task-**bundle**-based two-stage incentive mechanism for mobile crowd **sensing** (BundleSense) in detail. Then we analyze the theoretical properties of BundleSense.

A. Stage I

In Stage I, the winners in each bundle and each bundle's budget will be determined by the agent. The key challenge here is how to maximize the agent's budget for recruiting users. As maximizing the agent's budget can easily lead to untruthful bids, existing works mainly adopt relatively random algorithms, such as the SAMU algorithm in [11] and the SUCP algorithm in [19]. Here, we propose the algorithm AucB, which tries to maximize the budget for each agent strategically, and at the same time, assure the truthfulness of requesters.

Algorithm 1 AucB: **Auc**tion for determining budget and winners for all **B**undles

Input: Each bundle's set of tasks.
Output: Each agent's budget b_j.

for $j = 1$ to n **do**
  $\mathbf{t} \leftarrow [t_j^1, t_j^2, ..., t_j^{n_j}]$; // having $b_j^1 \geq b_j^2 \geq ... \geq b_j^{n_j}$
  Partition $\mathbf{t}$ uniformly at random into vectors $\mathbf{t}'$ and $\mathbf{t}''$;
  Let $\mathbf{b}'$ and $\mathbf{b}''$ be the budget vectors of $\mathbf{t}'$ and $\mathbf{t}''$;
  $F' \leftarrow \Phi(\mathbf{b}')$; $F'' \leftarrow \Phi(\mathbf{b}'')$;
  if $F' \leq F''$ **then**
    $(b_j, \mathcal{W}_j) \leftarrow (F', \text{ExtT}(F', \mathbf{b}''))$;
  else
    $(b_j, \mathcal{W}_j) \leftarrow (F'', \text{ExtT}(F'', \mathbf{b}'))$;
  end if
end for

Algorithm 1 demonstrates the auction AucB of BundleSense. For each bundle, tasks are sorted in nonincreasing order according to their budgets. The sorted vector $\mathbf{t}$ then is partitioned uniformly into two parts, $\mathbf{t}'$ and $\mathbf{t}''$. Specifically, each task $t_j^k \in \mathbf{t}$ has a probability 1/2 to be put into $\mathbf{t}'$ and otherwise $\mathbf{t}''$. Let $\mathbf{b}'$ and $\mathbf{b}''$ be their corresponding budget vectors. The optimal single-price auction [21] is adopted to calculate the optimal budget F' and F'' for $\mathbf{b}'$ and $\mathbf{b}''$, respectively. In [21], the profit of the optimal single-price auction $\widetilde{\Phi}(\mathbf{b})$ considering a nonincreasing vector $\mathbf{b}$ is determined by:

$$\widetilde{\Phi}(\mathbf{b}) = \max_{1 \leq j \leq |\mathbf{b}|} j b_j. \tag{6}$$

Since we take into account the time delay caused by bundling, we rewrite the profit as:

$$\Phi(\mathbf{b}) = \max_{1 \leq j \leq |\mathbf{b}|} j b_j \lambda(j). \tag{7}$$

Then we compute $F' = \Phi(\mathbf{b}')$ and $F'' = \Phi(\mathbf{b}'')$.

If $F' \leq F''$, the tasks in $\mathbf{t}''$ (with the budgets in $\mathbf{b}''$) form the candidate set and winners will be selected from them. The tasks in $\mathbf{t}'$ form the reference set, and the corresponding value of F' is used as the targeted budget in ExtT (Algorithm 2). If $F' > F''$, $\mathbf{t}'$ becomes the candidate set of tasks, $\mathbf{t}''$ becomes the reference and F'' is the targeted budget. As will be discussed later, the segmentation of the candidate set and the reference set helps to guarantee not only the truthfulness of our algorithm but also the competitiveness. In ExtT, the largest j^* is found in the sorted candidates' budget vector $\mathbf{d}$ such that the highest j^* budgets of tasks are at least $\frac{R}{j^* \lambda(j^*)}$. Then these j^* tasks become the winners of the bundle $\mathcal{B}_j$. Note that in ExtT, the input budget vector $\mathbf{d}$ is part of the budget vector $\mathbf{b}$ in AucB. Therefore, d_k in ExtT is not necessarily equal to b_j^k. Thus in Line 2 of ExtT, we use $t_j^{(k)}$ to denote the task in $\mathcal{B}_j$ corresponding to the budget $d_k \in \mathbf{d}$.

Algorithm 2 ExtT: **Ext**racting winning **T**asks under target R

Input: The targeted budget R, the sorted budget vector $\mathbf{d}$.
Output: The set of winners $\mathcal{W}$.

Find the largest $j^* \in [1, 2, ..., |\mathbf{d}|]$, such that $j^* d_{j^*} \lambda(j^*) \geq R$;
$\mathcal{W} \leftarrow \{t_j^{(k)} | k \leq j^*\}$;

Algorithm 3 AucAU: **Auc**tion for matching **A**gents and **U**sers

Input: Agents' budget vector $\mathbf{b} = [b_1, b_2, ..., b_n]$ in descending order, users' bid vector $\mathbf{s} = [s_1, s_2 ..., s_m]$ in ascending order.
Output: A matching result between agents and users Λ and the corresponding payments and rewards.

1: Let k denote the largest index that $b_k \geq s_k$;
2: $\mathbf{b}' = [b_1, b_2, ..., b_k]$; $\mathbf{s}' = [s_1, s_2, ..., s_k]$;
3: **if** $k = 2$ **then**
4:     Run the Vickery auction, output its result, and halt.
5: **end if**
6: $\mathbf{b}'' \leftarrow [b_1 - s_k, b_2 - s_k, ..., b_{k-1} - s_k]$;
7: $\mathbf{s}'' \leftarrow [b_k - s_1, b_k - s_2, ..., b_k - s_{k-1}]$;
8: Let $0 \leq X \leq 1$ be a uniformly random value;
9: **if** $X < 0.5$ **then**
10:     $(\mathcal{W}, \rho) \leftarrow$AucSP($\mathbf{b}''$);
11:     Agents in $\mathcal{W}$ win at $p_j = \rho + s_k$;
12:     $(\mathcal{W}', \rho') \leftarrow$kVickrey($\mathbf{s}', |\mathcal{W}|$);
13:     Users in $\mathcal{W}'$ win at $r_i = \rho'$;
14: **else**
15:     $(\mathcal{W}', \rho') \leftarrow$AucSP($\mathbf{s}''$);
16:     Users in $\mathcal{W}'$ win at $r_i = b_k - \rho'$;
17:     $(\mathcal{W}, \rho) \leftarrow$kVickrey($\mathbf{b}', |\mathcal{W}'|$);
18:     Agents in $\mathcal{W}$ win at $p_j = \rho$;
19: **end if**
20: $\Lambda = \{(a_1, u_1), (a_2, u_2), ..., (a_{|\mathcal{W}|}, u_{|\mathcal{W}|})\}$;

B. Stage II

After determining the winners and budget of each bundle, a double auction is performed between users and agents in Stage II of BundleSense. Existing incentive mechanisms are mostly applicable to scenarios where users bid unilaterally, given that the task's budget is fixed. Differently, agents and users bid at the same time in our model. The challenge here is how to maximize the utility of the platform while ensuring the truthfulness of both agents and users. In addition, it is also important to ensure a constant competitive ratio between the proposed mechanism and the optimal mechanism.

Algorithm 3 sketches the proposed auction for matching agents and users (AucAU). At the beginning of AucAU, we sort all the budgets of agents in descending order (recorded as the vector $\mathbf{b}$) and all the bids of users in ascending order (recorded as the vector $\mathbf{s}$). We then find the largest k such that $b_k \geq s_k$, reserving the highest k budgets and the lowest k bids. If $k = 2$, we run the Vickery auction in $\mathbf{b}$ and $\mathbf{s}$, respectively. The Vickery auction selects the agent a_1 whose payment will be b_2 and the user u_1 will get a reward s_2. If $k > 2$, we generate two new vectors $\mathbf{b}'' = [b_j - s_k | 1 \leq j \leq k-1]$ and $\mathbf{s}'' = [b_k - s_i | 1 \leq i \leq k-1]$. Both $\mathbf{b}''$ and $\mathbf{s}''$ are in descending order and have $k-1$ elements, which are the key to ensure that the platform is budget balanced. Then we run different mechanisms strategically based on $\mathbf{b}''$ and $\mathbf{s}''$.

Algorithm 4 AucSP: **Auc**tion for determining winning **S**et and winning **P**rice.

Input: The vector in descending order $\mathbf{b} = \{b_1, b_2, ..., b_n\}$
Output: The set of winners $\mathcal{W}$ and the winning price ρ

Partition $\mathbf{b}$ uniformly at random into two vectors, for each budget, with probability 1/2 put the budget in $\mathbf{b}'$ and otherwise $\mathbf{b}''$.
Let ω', ω'' be the corresponding vectors of agents or users;
$F' \leftarrow \widetilde{\Phi}(\mathbf{b}')$; $F'' \leftarrow \widetilde{\Phi}(\mathbf{b}'')$;
if $F' \leq F''$ **then**
    $(\mathcal{W}, \rho) \leftarrow$ExtSP($F', \mathbf{b}'', \omega''$);
else if $F'' < F'$ **then**
    $(\mathcal{W}, \rho) \leftarrow$ExtSP($F'', \mathbf{b}', \omega'$);
end if

Algorithm 5 ExtSP: **Ext**racting the best winner **S**et and winning **P**rice under target R

Input: The targeted budget R, the sorted budget vector $\mathbf{d}$ and the corresponding set of agents or users ω.
Output: The best winner set $\mathcal{W}$ and the winning price ρ.

Find the largest $k^* \in [1, 2, ..., |\omega|]$ such that $k^* d_{k^*} \geq R$;
$\mathcal{W} \leftarrow \{\omega_1, ..., \omega_{k^*}\}$; $\rho \leftarrow R/k^*$;

Specifically, a random value X is generated first in Line 8 of AucAU, which determines the execution of different strategies. If $X < 0.5$, we run AucSP (Algorithm 4) in $\mathbf{b}''$, obtaining the winner set of agents $\mathcal{W}$ and the winning payment $p_j = \rho + s_k$. Then, we run the k-Vickrey auction, which selects $|\mathcal{W}|$ users with the reward $r_i = r_{|\mathcal{W}|+1}$. If $X > 0.5$, we run AucSP in $\mathbf{s}''$, obtaining the winning users and the winning reward. We then select the corresponding number of agents to match these users. At the end of the algorithm, we can simply match each selected agent with one selected user randomly from the winning sets. Each winning task (determined at Stage I) in the selected bundles will be charged a payment $p_j^k = \frac{b_j}{w_j}$, where w_j is the number of winning tasks in the bundle $\mathcal{B}_j$.

Note that the function of AucSP is very similar to AucB (Algorithm 1). The difference is that, in AucSP, we use the original optimal single-price auction $\widetilde{\Phi}(\mathbf{b})$ to replace $\Phi(\mathbf{b})$ and use ExtSP (Algorithm 5) to replace ExtT in AucB. The design of AucSP helps the proposed double auction algorithm AucAU to suit the situation with two types of bidders. The functions of ExtSP and ExtT are also similar. However, ExtSP does not need to consider the function $\lambda(\cdot)$ and it returns the winners and the winning price.

C. Mechanism Analysis

1) The Analysis of Stage I:

Lemma 1. *AucB is computationally efficient.*

Proof: In AucB, sorting $\mathbf{t}$ needs $O(n_j \log_2(n_j))$ time. Since the value of n_j is less than M, this process is bounded by $O(M \log_2(M))$. Partitioning $\mathbf{t}$ into two sets and the calculation of Φ take $O(M)$ time. In ExtT, the largest j^* can be found in $O(M)$ time. Since the for-loop of AucB (Line 1-Line 11) goes through all the bundles, AucB takes at most $O(nM \log_2(M))$ time. ■

Lemma 2. *AucB is individually rational.*

Proof: The task t_j^k's payment p_j^k is zero if t_j^k loses at Stage I or his agent loses the auction at Stage II. Otherwise, if $t_j^k \in \mathcal{W}_j$ and $a_j \in \Lambda_a$, the task will be charged a payment of $p_j^k = b_j/w_j$, where b_j is equal to the targeted budget R in ExtT, w_j is the largest index in $\mathbf{d}$ of ExtT satisfying $d_{w_j} w_j \lambda(w_j) \geq R$ (Line 1 in ExtT). Since $\mathbf{d}$ is in descending order, each winning task t_j^k satisfies $b_j^k \geq d_{w_j}$. Thus, we have

$$p_j^k = b_j/w_j = R/w_j \leq d_{w_j}\lambda(w_j) \leq b_j^k \lambda(w_j). \quad (8)$$

Given that the requester reports his true valuation, i.e., $b_j^k = v_j^k$, we have $v_j^k \lambda(w_j) - p_j^k \geq 0$, which completes the proof. ■

Lemma 3. *AucB is truthful.*

The proof of Lemma 3 is given in Appendix-A.

Lemma 4. *AucB is 4-competitive.*

The proof of Lemma 4 is given in Appendix-B. With Lemma 4, We can show the advantage of AucB compared with SAMU [11] and SUCP [19]. Specifically, assume that the budget of tasks in the bundle $\mathcal{B}_j$ obeys an uneven distribution:

$$b_j^k = \begin{cases} n_j, & k = 1, 2, \\ 1, & 3 \leq k \leq n_j, \end{cases} \quad (9)$$

where $n_j \geq 3$. Let $\mathbf{b} = \{b_j^k | 1 \leq k \leq n_j\}$, then we have $\Phi(\mathbf{b}) = 2n_j\lambda(2)$. According to Lemma 4, AucB$(\mathbf{b}) \geq \frac{n_j \lambda(2)}{2}$. According to SUCP, a_j randomly chooses $k' \in [1, n_j]$ and uses $b_j^{k'}$ as the clearing price, selecting tasks whose budgets are higher than $b_j^{k'}$. If there are ω selected tasks, SUCP's budget will be $b_j^{k'} \omega \lambda(\omega)$. When $k' = 1$ or $k' = 2$, no task's budget is higher than n_j and $b_j = 0$. Otherwise when $k' \geq 3$, t_j^1 and t_j^2 will be winning tasks and $b_j = 2\lambda(2)$. Therefore, the expected budget of a_j is $P[k' = 1 | k' = 2] * 0 + P[k' \geq 3] * 2\lambda(2) = \frac{2}{n_j} * 0 + \frac{n_j - 2}{n_j} * 2\lambda(2) = (2 - \frac{4}{n_j})\lambda(2)$. Thus, the ratio of the expected budget achieved by AucB to that by SUCP is $\frac{\frac{n_j}{2}}{2 - \frac{4}{n_j}} > \frac{n_j}{4}$. Similarly, the ratio between AucB and SAMU is also $\Omega(n_j)$, which shows that AucB is more competitive.

Theorem 1. *AucB is computationally efficient, individually rational, truthful and constantly competitive.*

2) The Analysis of Stage II:

Lemma 5. *AucAU is computationally efficient.*

Proof: First, sorting the two input vectors is bounded by $O(\max\{n, m\} \log_2(\max\{n, m\}))$ time. Converting $\mathbf{b}$ and $\mathbf{s}$ into the same length requires $O(\min\{m, n\})$ time. Then we analyze the algorithm AucSP and kVickrey. The difference between AucSP and AucB is that AucSP does not need to perform sorting and it does not contain for-loop. Thus, AucSP is bounded by $O(\min\{m, n\})$ time. kVickrey can be done in $O(1)$ time. Combining all these steps, it can be concluded that the time complexity of AucAU is $O(\max\{m, n\} \log_2(\max\{m, n\})$. ■

Lemma 6. *AucAU is individually rational.*

Proof: Agents' or users' utility will be zero if they lose the auction. Therefore, we only need to consider the case when they win. In Algorithm 5, it can be seen that k^* is the largest index such that $k^* d_{k^*} \geq R$ and d_{k^*} is the lowest winning price. If $X < 0.5$, AucSP($\mathbf{b}''$) will be executed and each winning agent a_j has $b_j'' \geq d_{k^*} \geq R/k^*$. His payment is $p_j = \rho + s_k = R/k^* + s_k \leq b_j'' + s_k$, where $b_j'' = b_j - s_k$. Then we have $p_j \leq b_j$. If $X \geq 0.5$, a_j's payment will be $b'_{|\mathcal{W}'|+1}$. Because $\mathbf{b}'$ is in decreasing order, this payment is also less than his budget. Combining these two cases, AucAU satisfies individual rationality for agents. The same proof process applies for the individual rationality for users and thus is omitted here. ■

Lemma 7. *AucAU is budget balanced for the platform.*

Proof: If $X < 0.5$, AucSP($\mathbf{b}''$) will be executed and $|\mathcal{W}|$ agents will be selected. For $|\mathbf{b}''| = k-1$, we have $|\mathcal{W}| < k-1$. In Line 11 of AucAU, each winning agent a_j pays an amount of $p_j = \rho + s_k$, where ρ is a positive value and thus $p_j > s_k$. Then each winning user gets a reward $r_i = s'_{|\mathcal{W}|+1}$. Because $|\mathcal{W}| + 1 < k$ and $\mathbf{s}$ is in increasing order, each reward $r_i < s_k$, namely for any agent-user pair we have $p_j > s_k > r_i$, which shows that the platform is budget balanced. The same proof process applies for the case when $X \geq 0.5$ and thus is omitted here. ■

Lemma 8. *AucAU is double truthful for agents and users.*

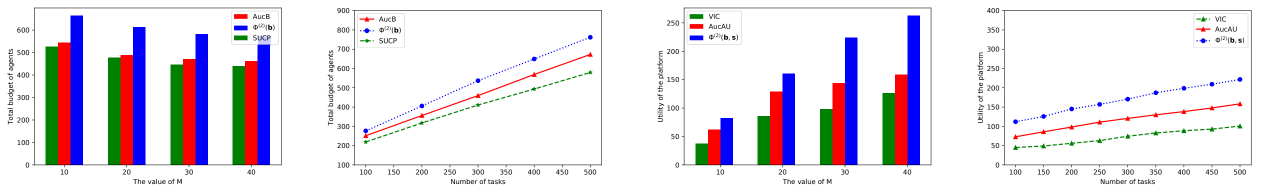
Proof: In Lemma 3 we have proven that AucB is truthful. Since AucSP adopts the same framework as AucB, it can be concluded that AucSP is also truthful. AucAU is a combination of the truthful Vickery auction [22] and AucSP in terms of probability. Thus AucAU is also truthful. ■

Lemma 9. *AucAU is 8-competitive.*

The proof of Lemma 9 is in Appendix C.

Theorem 2. *AucAU is computationally efficient, individually rational, budget balanced, double truthful and constantly competitive.*

3) The Analysis of BundleSense: As BundleSense is a sequential combination of AucB and AucAU, it can be directly inferred that BundleSense satisfies the properties of computational efficiency, individual rationality, budget balance, and constant competitiveness according to Theorem 1 and Theorem 2. BundleSense also achieves triple truthfulness with respect to requesters, agents, and users by these two theorems.



(a) Agent's budget by varying M. (b) Agent's budget by varying the number of tasks. (c) Platform's utility by varying M. (d) Platform's utility by varying number of tasks.

Fig. 2. Performance comparison.

Theorem 3. *BundleSense is computationally efficient, individually rational, budget balanced, triple truthful and constantly competitive.*

V. Performance Evaluation

A. Simulation Setup

Locations of users and tasks are distributed in a 2D data space $[0, 500]^2$. We generate 30 users $(m = 30)$ and 200 tasks $(\sum_j n_j = 200)$ in total, varying the maximum bundle capacity M from 10 to 40 with an increment of 10. The number of agents (n) is determined by the result of bundling. In stage I, each task t_j^k's valuation v_j^k and budget b_j^k are uniformly distributed within $[5, 10]$. In stage II, each user's reserved price s_i and bid v_i are uniformly distributed in $[20, 30]$, which are higher than the budget of any individual task.

B. Evaluation Results

Fig. 2(a)-(b) compare the budgets collected by agents computed by AucB, $\Phi^{(2)}(\mathbf{b})$ (the optimal auction introduced in Appendix-B) and SUCP [19], respectively. SUCP randomly chooses one task t_j^k and uses b_j^k as the clearing price, selecting all tasks with budgets higher than b_j^k. If there are κ winning tasks, the budget for a_j will be $b_j^k \kappa\lambda(\kappa)$. It can be seen that the optimal $\Phi^{(2)}(\mathbf{b})$ always obtains the highest budget compared to the other algorithms and the proposed AucB outperforms SUCP. The biggest competitive ratio between AucB and $\Phi^{(2)}(\mathbf{b})$ in Fig. 2(a) is 1.22 when $M = 10$ and 1.17 in Fig. 2(b) when there are 500 tasks. As M increases or the number of users decreases, the ratio further decreases. The results show that the competitive ratio between AucB and the optimal solution is better in practice than the theoretical result.

Fig. 2(c)-(d) compare the utility of the platform calculated by AucAU, $\Phi^{(2)}(\mathbf{b}, \mathbf{s})$ (the optimal auction introduced in Appendix-C) and VIC, respectively. VIC is a truthful mechanism adapted from the basic Vickery mechanism in the double auction with random selection. Given a nonincreasing budget vector $\mathbf{b}$ and a nondecreasing bid vector $\mathbf{s}$, VIC finds the largest k^* such that $b_{k^*} \geq s_{k^*}$ and then chooses $m \in [1, 2, ..., k^* - 1]$ randomly. The first m agents and m users win with the payment b_{m+1} and the reward s_{m+1}, respectively. It can be seen that $\Phi^{(2)}(\mathbf{b}, \mathbf{s})$ outperforms the other mechanisms and AucAU performs better than VIC. In Fig. 2(c), when $M = 40$, the gap between AucAU and $\Phi^{(2)}(\mathbf{b}, \mathbf{s})$ is maximal, with the competitive ratio of 1.71. In Fig. 2(d), the maximal competitive ratio is 1.54 when there are 100 tasks and the ratio decreases when there are more tasks. Again, the competitive ratio of the proposed AucAU to the optimal solution in the experiment is much better than the theoretical result.

VI. Conclusion

In this paper, we have proposed a task-bundling-based two-stage incentive mechanism for MCS. The proposed mechanism can effectively bridge the gap between task requesters with low recruitment budgets and mobile users with relatively high working prices, and thus improve the applicability of MCS systems. Through theoretical analysis and simulations, the proposed mechanism has been proved that possess the desirable properties of computational efficiency, individual rationality, budget balance, truthfulness, and constant competitiveness.

Appendix

A. Proof of Lemma 3

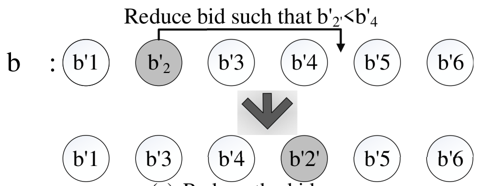
We first prove that the function $k\lambda(k)$ in Algorithm 1 and Algorithm 2 is monotonically increasing. It can be derived that the function $k\lambda(k)$ is monotonically increasing when $k \leq M + 1/2$. In Algorithm 1, we uniformly divide the vector $\mathbf{b}$ into two parts $\mathbf{b}'$ and $\mathbf{b}''$ at random and we have $|\mathbf{b}'| < M$ and $|\mathbf{b}''| < M$, namely $k\lambda(k)$ in our algorithm is monotonically increasing.

Then we are ready to discuss four cases to show that the task requester cannot improve his utility by submitting an untruthful value.

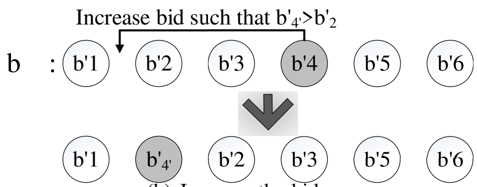
Case 1: The requester of t_j^k fails no matter he bids truthfully or untruthfully. Here the requester's utility is always zero.

Case 2: The requester of t_j^k wins when he bids truthfully and fails when he bids untruthfully. By the designed rule, the winner's utility is nonnegative and the loser's utility is zero. Therefore, the requester should bid truthfully here.

Case 3: The requester of t_j^k fails when he bids truthfully and wins when he bids untruthfully. By AucB, when the requester bids truthfully, the first situation of failure is that his bid is in $\mathbf{b}'$ while $F' \leq F''$ or his bid is in $\mathbf{b}''$ while $F'' < F'$. In other words, t_j^k is in the reference set, but not in the candidate set for selection. Without loss of generality, assume that $F' \leq F''$ and the truthful bid for t_j^k is the κ-th member of $\mathbf{b}'$, namely $b'_\kappa = v_j^k$. The requester's only way to improve utility is reporting an



(a) Reduce the bid.



(b) Increase the bid.

Fig. 3. Two ways to change bids for t_j^k.

untruthful bid $b'_{\kappa'}$, such that $F' > F''$. Then $\text{ExtT}(F'', \mathbf{b}')$ will be executed and t_j^k will be in the candidate set. The requester can take the following actions: (i) Reduce the bid b'_κ to $b'_{\kappa'}$ like changing b'_2 to $b'_{2'}$ in Fig. 3(a). The budgets for entries 2-4 are all reduced (b'_2 is reduced to b'_3, b'_3 is reduced to b'_4 and b'_4 is reduced to $b'_{2'}$). Thus F' cannot be increased and the requester cannot improve his utility in this way. (ii) Increase the bid b'_κ to $b'_{\kappa'}$ like changing b'_4 to $b'_{4'}$ in Fig. 3(b). The budgets for entries 2-4 are all increased and F' may be increased. If the requester's untruthful bid makes $F' > F''$ successfully, then there will exist no more than four winners ($b'_5 5\lambda(5)$ is still less than F'' and he cannot be a winner). Let w_j be the number of winners. t_j^k's payment satisfies $p_j^k = F''/w_j$ and the corresponding utility $\mu_j^k = b'_4\lambda(w_j) - F''/w_j$. In the original nonincreasing sequence $\mathbf{b}'$, for any $w_j \leq 4$, we have $F'' \geq b'_{w_j} w_j \lambda(w_j) \geq b'_4 w_j \lambda(w_j)$. Thus $b'_4\lambda(w_j) \leq F''/w_j$. In other words, the utility is always nonpositive.

The second situation of failure is that, t_j^k is in the candidate set for selection, but its budget fails the auction in ExtT. We also assume that the bid for t_j^k is the κ-th entry of $\mathbf{b}'$. In this case, we have $F' > F''$ and $\kappa \geq j^* + 1$, where j^* is the largest index in ExtT satisfying $j^* d_{j^*} \lambda(j^*) \geq R = F''$. We consider two actions: (i) Reduce his bid. The requester cannot win by this mean and winners are always the first j^* tasks. (ii) Increase the bid b'_κ to $b'_{\kappa'}$ like changing b'_4 to $b'_{4'}$ in Fig. 3(b). In such case, there will not have more than κ winners. Let $w_j \leq \kappa$ be the number of winners, μ_j^k will be $b'_\kappa\lambda(w_j) - F''/w_j$. At the start of Appendix A we proved that $k\lambda(k)$ in our mechanism is always increasing. Thus, for any $w_j \leq \kappa$, we have $F'' > b'_\kappa \kappa\lambda(\kappa) \geq b'_\kappa w_j \lambda(w_j)$. Hence μ_j^k is always a negative value.

Case 4: The requester of t_j^k wins no matter he bids truthfully or untruthfully. Assume that the bid for t_j^k is the κ-th entry of $\mathbf{b}'$. In this case, we have $F' > F''$ and $\kappa \leq j^*$. The requester's utility μ_j^k will be $b'_\kappa\lambda(w_j) - F''/w_j$, where w_j is the number of winners. Note that F'' is calculated by the reference set and cannot be influenced by t_j^k. In addition, a winning untruthful bidding cannot change the number of winners w_j either. The reason is that: Assume in Fig. 3(b), $w_j = 4$ and $v_j^k = b'_2$. If his untruthful bid $b'_{2'} \geq b'_4$, then the largest j^* in ExtT is always four because $b'_4 4\lambda(4) \geq F''$. If $b'_{2'} < b'_4$, any winning untruthful bid should satisfies $b'_{2'} > b'_5$, otherwise $b'_{2'} 5\lambda(5) < b'_5 5\lambda(5) < F''$ and he will lose. Thus $b'_5 < b'_{2'} < b'_4$ and $b'_{2'} 4\lambda(4) \geq F''$, namely there are also four winners. In summary, the requester cannot improve his utility in this case.

Combining these four cases, we complete our proof.

B. Proof of Lemma 4

In our discussions to follow, it will be useful to compare the performance of truthful auctions to that of the optimal single price omniscient auction (Eq. (7)). Lemma 3.5 in [21] has shown that no truthful auction can be competitive against Φ. Thus, we only focus on the scenarios where there are at least two participants, which are common for MCS systems. The following lemma can be directly derived from [21].

Lemma 10. *Given an input vector* $\mathbf{b}$*, the optimal single price omniscient auction* $\Phi^{(2)}$ *that sells at least two items from* $\mathbf{b}$ *by considering* $\lambda(\cdot)$ *is defined by:*

$$\Phi^{(2)}(\mathbf{b}) = \max_{2 \leq j \leq |\mathbf{b}|} j b_j \lambda(j). \quad (10)$$

By the lemma, $\Phi^{(2)}$ on $\mathbf{b}$ finds $j \geq 2$ tasks at price p and $\Phi^{(2)}(\mathbf{b}) = jp\lambda(j)$ in our scenario. These j tasks, all with a budget value at least p, are divided uniformly at random into $\mathbf{b}'$ and $\mathbf{b}''$. Let j' be the number of tasks in $\mathbf{b}'$ and j'' be the number of tasks in $\mathbf{b}''$. Since $\Phi(\mathbf{b}')$ and $\Phi(\mathbf{b}'')$ are the best possible results on $\mathbf{b}'$ and $\mathbf{b}''$, respectively, we have $\Phi(b') \geq pj'\lambda(j')$ and $\Phi(b'') \geq pj''\lambda(j'')$. Therefore

$$\frac{B}{\Phi^{(2)}(\mathbf{b})} = \frac{\min\{\Phi(\mathbf{b}'), \Phi(\mathbf{b}'')\}}{\Phi^{(2)}(\mathbf{b})} \geq \frac{\min\{j'\lambda(j'), j''\lambda(j'')\}}{j\lambda(j)}.$$

Through the inequality above, we get the competitive ratio:

$$\frac{\mathbf{E}[B]}{\Phi^{(2)}(\mathbf{b})} \geq \sum_{i=1}^{j-1} \frac{\min\{i\lambda(i), (j-i)\lambda(j-i)\}}{j\lambda(j)} \binom{j}{i} 2^{-j}.$$

Here $\binom{j}{i}2^{-j}$ is the possibility that $j' = i$ and $\frac{\min\{i\lambda(i), (j-i)\lambda(j-i)\}}{j\lambda(j)}$ is the corresponding competitive ratio. Adding up all the possible cases generates the final result.

Next, we analyze the impact of $\lambda(\cdot)$ on the competitive ratio. We have

$$\frac{\min\{j'\lambda(j'), j''\lambda(j'')\}}{j\lambda(j)} \geq \frac{\min\{j', j''\}\min\{\lambda(j'), \lambda(j'')\}}{j\lambda(j)}.$$

Compared to competitive ratio $\frac{\min\{j', j''\}}{j}$ of RSPE [21], we add a term greater than or equal to $\frac{\min\{\lambda(j'), \lambda(j'')\}}{\lambda(j)}$. For $j > \min\{j', j''\}$ and the function $\lambda(\cdot)$ is decreasing. Thus we have $\min\{\lambda(j'), \lambda(j'')\}/\lambda(j) > 1$. In other words, by introducing $\lambda(\cdot)$, we not only quantify the impact of the time delay caused by bundling on the actual utility of the task but also decrease the competitive ratio of auctions.

To summarize, when $M = 1$, it is equivalent to no bundling and the competitive ratio is equal to RSPE's ratio of four. For any $k \geq 2$, the ratio achieves its maximum of four when

$k=2$ and $M \to \infty$. As k increases or M decreases, the ratio decreases gradually.

C. Proof of Lemma 9

Similar to the proof of Lemma 4, we consider the competitiveness of AucAU compared with the optimal single price omniscient mechanism that transfers at least two items [16].

Lemma 11. *Given input vectors* $\mathbf{b}$ *and* $\mathbf{s}$*, the optimal single price omniscient mechanism for double auction* Φ *that transfers at least two items is defined by:*

$$\Phi^{(2)}(\mathbf{b},\mathbf{s}) = \max_{2\le i\le\varepsilon} i(b_i - s_i), \tag{11}$$

where $\varepsilon = \min\{|\mathbf{b}|, |\mathbf{s}|\}$.

At the beginning of AucAU, we find the largest k satisfying $b_k \ge s_k$. If $k=2$, AucAU runs Vickrey and is 2-competitive. If $k \ge 3$, let $\tau \ge 2$ be the number of agent-user pairs selected by $\Phi^{(2)}(\mathbf{b},\mathbf{s})$. Thus,

$$\begin{aligned}\Phi^{(2)}(\mathbf{b},\mathbf{s}) &= \max_{2\le\tau\le k} \tau(b_\tau - s_\tau)\\ &= \tau(b_k - s_\tau) + \tau(b_\tau - s_k) - \tau(b_k - s_k).\end{aligned}$$

Let $\mathbf{b}^* = [b_j - s_k | 1 \le j \le k]$ and $\mathbf{s}^* = [b_k - s_i | 1 \le i \le k]$. Let $\widetilde{\Phi}^{(2)}(\mathbf{b}) = \max_{2\le j\le|\mathbf{b}|} j b_j$ denotes the optimal single price omniscient auction that sells at least two items. Since $\widetilde{\Phi}^{(2)}(\mathbf{b})$ is the best possible results on $\mathbf{b}$, we have $\widetilde{\Phi}^{(2)}(\mathbf{b}^*) \ge \tau(b_\tau - s_k)$ and $\widetilde{\Phi}^{(2)}(\mathbf{s}^*) \ge \tau(b_k - s_\tau)$. Thus

$$\Phi^{(2)}(\mathbf{b},\mathbf{s}) \le \widetilde{\Phi}^{(2)}(\mathbf{b}^*) + \widetilde{\Phi}^{(2)}(\mathbf{s}^*) - \tau(b_k - s_k).$$

Let $\mathbf{b}''$ (resp. $\mathbf{s}''$) be $\mathbf{b}^*$ with the lowest budget (resp. the highest bid) deleted. Namely $\mathbf{b}'' = [b_1 - s_k, b_2 - s_k, ..., b_{k-1} - s_k]$ and $\mathbf{s}'' = [b_k - s_1, b_k - s_2, ..., b_k - s_{k-1}]$. Then

$$\widetilde{\Phi}^{(2)}(\mathbf{b}^*) \le \widetilde{\Phi}^{(2)}(\mathbf{b}'') + (b_k - s_k).$$

We have

$$\Phi^{(2)}(\mathbf{b},\mathbf{s}) \le \widetilde{\Phi}^{(2)}(\mathbf{b}'') + \widetilde{\Phi}^{(2)}(\mathbf{s}'') - (\tau - 2)(b_k - s_k).$$

Because $\tau \ge 2$ and $b_k \ge s_k$, then

$$\Phi^{(2)}(\mathbf{b},\mathbf{s}) \le \widetilde{\Phi}^{(2)}(\mathbf{b}'') + \widetilde{\Phi}^{(2)}(\mathbf{s}'').$$

AucAU runs AucSP($\mathbf{b}''$) if $X < 0.5$ otherwise runs AucSP($\mathbf{s}''$). According to Lemma 4, AucB is 4-competitive. The difference between AucB and AucSP is the use of $\lambda(\cdot)$. Without $\lambda(\cdot)$, AucSP($\mathbf{b}$) is 4-competitive with $\widetilde{\Phi}^{(2)}(\mathbf{b})$. Thus the expected utility from cases $X < 0.5$ and $X \ge 0.5$ are $0.5 * \widetilde{\Phi}^{(2)}(\mathbf{b}'')/4$ and $0.5 * \widetilde{\Phi}^{(2)}(\mathbf{s}'')/4$ respectively. Thus

$$\mathbb{E}[\text{AucAU}(\mathbf{b},\mathbf{s})] \ge \frac{1}{8}(\widetilde{\Phi}^{(2)}(\mathbf{b}'') + \widetilde{\Phi}^{(2)}(\mathbf{s}'')) \ge \Phi^{(2)}(\mathbf{b},\mathbf{s})/8,$$

which completes our proof.

References

[1] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2419–2465, 2019.

[2] B. Pan, Y. Zheng, D. Wilkie, and C. Shahabi, "Crowd sensing of traffic anomalies based on human mobility and social media," in *ACM SIGSPATIAL*, 2013, pp. 344–353.

[3] Y. Jing, B. Guo, Z. Wang, V. O. Li, J. C. Lam, and Z. Yu, "Crowdtracker: Optimized urban moving object tracking using mobile crowd sensing," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3452–3463, 2017.

[4] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: an end-to-end participatory urban noise mapping system," in *ACM/IEEE IPSN*, 2010, pp. 105–116.

[5] H. Wang, S. Guo, J. Cao, and M. Guo, "Melody: A long-term dynamic quality-aware incentive mechanism for crowdsourcing," in *IEEE ICDCS*, 2017, pp. 933–943.

[6] G. Ji, B. Zhang, Z. Yao, and C. Li, "A reverse auction based incentive mechanism for mobile crowdsensing," in *IEEE ICC*, 2019, pp. 1–6.

[7] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, "Robust incentive tree design for mobile crowdsensing," in *IEEE ICDCS*, 2017, pp. 458–468.

[8] J. Cui, Y. Sun, H. Huang, H. Guo, Y. Du, W. Yang, and M. Li, "Tcam: A truthful combinatorial auction mechanism for crowdsourcing systems," in *IEEE WCNC*, 2018, pp. 1–6.

[9] J. Xu, C. Guan, H. Wu, D. Yang, L. Xu, and T. Li, "Online incentive mechanism for mobile crowdsourcing based on two-tiered social crowdsourcing architecture," in *IEEE SECON*, 2018, pp. 1–9.

[10] Z. Wang, J. Li, J. Hu, J. Ren, Z. Li, and Y. Li, "Towards privacy-preserving incentive for mobile crowdsensing under an untrusted platform," in *IEEE INFOCOM*, 2019, pp. 2053–2061.

[11] P. Lin, X. Feng, Q. Zhang, and M. Hamdi, "Groupon in the air: A three-stage auction framework for spectrum group-buying," in *IEEE INFOCOM*, 2013, pp. 2013–2021.

[12] D. Yang, G. Xue, and X. Zhang, "Group buying spectrum auctions in cognitive radio networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 810–817, 2017.

[13] H. Xie and J. C. S. Lui, "Incentive mechanism and rating system design for crowdsourcing systems: Analysis, tradeoffs and inference," *IEEE Transactions on Services Computing*, vol. 11, no. 1, pp. 90–102, 2018.

[14] Z. Wang, J. Hu, Q. Wang, R. Lv, J. Wei, H. Chen, and X. Niu, "Task-bundling-based incentive for location-dependent mobile crowdsourcing," *IEEE Communications Magazine*, vol. 57, no. 2, pp. 54–59, 2019.

[15] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao, "Incentives for mobile crowd sensing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 54–67, 2016.

[16] K. Deshmukh, A. V. Goldberg, J. D. Hartline, and A. R. Karlin, "Truthful and competitive double auctions." *Lecture Notes in Computer Science*, vol. 2461, pp. 127–130, 2002.

[17] X. Zhai, T. Zhou, C. Zhu, B. Chen, W. Fang, and K. Zhu, "Truthful double auction for joint internet of energy and profit optimization in cognitive radio networks," *IEEE Access*, vol. 6, pp. 23 180–23 190, 2018.

[18] H. Zhang, B. Liu, H. Susanto, G. Xue, and T. Sun, "Incentive mechanism for proximity-based mobile crowd service systems," in *IEEE INFOCOM*, 2016, pp. 1–9.

[19] L. Huang, Y. Zhu, J. Yu, and M. Wu, "Group buying based incentive mechanism for mobile crowd sensing," in *IEEE SECON*, 2016, pp. 1–9.

[20] C. Peng, L. Xiang, C. Zhao, C. Lei, and J. Zhao, "Reliable diversity-based spatial crowdsourcing by moving workers," in *VLDB*, vol. 8, no. 10, 2015, pp. 1022–1033.

[21] A. V. Goldberg, J. D. Hartline, A. R. Karlin, M. Saks, and A. Wright, "Competitive auctions ," *Games & Economic Behavior*, vol. 55, no. 2, pp. 242–269, 2006.

[22] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *Journal of Finance*, vol. 16, no. 1, pp. 8–37, 1961.