Auction-Based Combinatorial Multi-Armed Bandit Mechanisms with Strategic Arms

Guoju Gao[†], He Huang[†*], Mingjun Xiao[‡*], Jie Wu[§], Yu-E Sun[¶], and Sheng Zhang[||]
[†]School of Computer Science and Technology, Soochow University, China
[‡]School of Computer Science and Technology, University of Science and Technology of China
[§]Department of Computer and Information Sciences, Temple University
[¶]School of Rail Transportation, Soochow University, China
[||]State Key Laboratory for Novel Software Technology, Nanjing University, China
[*]Correspondence to: huangh@suda.edu.cn, xiaomj@ustc.edu.cn

***Abstract*—The multi-armed bandit (MAB) model has been deeply studied to solve many online learning problems, such as rate allocation in communication networks, Ad recommendation in social networks, etc. In an MAB model, given N arms whose rewards are unknown in advance, the player selects exactly one arm in each round, and his goal is to maximize the cumulative rewards over a fixed horizon. In this paper, we study the budget-constrained auction-based combinatorial multi-armed bandit mechanism with strategic arms, where the player can select K $(< N)$ arms in a round and pulling each arm has a unique cost. In addition, each arm might strategically report its cost in the auction. To this end, we combine the upper confidence bound (UCB) with auction to define the UCB-based rewards and then devise an auction-based UCB algorithm (called AUCB). In each round, AUCB selects the top K arms according to the ratios of UCB-based rewards to bids and further determines the critical payment for each arm. For AUCB, we derive an upper bound on regret and prove the truthfulness, individual rationality, and computational efficiency. Extensive simulations show that the rewards achieved by AUCB are at least 12.49% higher than those of state-of-the-art algorithms.**

***Index Terms*—Combinatorial multi-armed bandits, auction mechanism, strategic cost, regret bound, truthfulness.**

I. Introduction

In recent years, the multi-armed bandit model [1], [2] has been widely used to solve problems where some parameters are unknown in advance, such as the rate allocation in wireless channels [3], user selection in crowdsensing [4], Ad recommendation in social networks [5], etc. Generally, there exists a dilemma between exploration and exploitation in the MAB model. The exploration means that the player (a.k.a. decision-maker) will select some sub-optimal arms to find the potentially optimal arm that may yield higher rewards in the future; the exploitation indicates that the player will choose the arms that performed best in the past. In the stochastic MAB model, the player can select exactly one arm from N arms at each round. If an arm is selected in a round, a random reward, which is independent and identically distributed (i.i.d.) over time, is obtained by the player. The player's objective is to maximize the cumulative rewards over a fixed horizon. Here, maximizing the total rewards is equivalent to minimizing the regret, which is defined as the difference between the cumulative rewards achieved in the optimal policy and the total rewards of the non-optimal policy. The optimal policy means that the player knows the expected rewards of all arms in advance and always selects the optimal arm in each round.

In the classic multi-armed bandit model, each player can select only one arm in a round and the model assumes that pulling every arm has the same cost. In contrast, a variant of the classic MAB model, called combinatorial multi-armed bandit (CMAB), has been recently proposed [6], [7], in which the player can select exactly K arms from N arm candidates in each round. After K arms are pulled, the random reward of each individual arm is observed by the player. The more complicated case is where pulling each arm has a unique cost and the player has a limited budget. Now, the player's goal is to maximize the cumulative achieved rewards under the budget constraint. Furthermore, the traditional MAB model simply assumes that all arms are feelingless machines. However, in many application models [8], [9], the arms are generally rational and selfish individuals. In other words, each arm might strategically report its cost to maximize its own payoff. The strategic behavior of each arm would hardly harm the player's performance. In order to ensure that each arm has no incentive to lie, the auction-based CMAB model is necessary.

In this paper, we focus on the auction-based combinatorial multi-armed bandit (ACMAB) problem, in which the player and arms are seen as buyer and sellers of arm-pulling opportunity in the auction, respectively. Each arm will claim its cost (called "bid") at the beginning of the auction. Then, the player selects K arms according to the bids and rewards. Here, the rewards of arms are unknown prior. In fact, the player needs not only to determine K arms (called winning arms) in each round, but also to calculate the payment for each winning arm. Note that each arm is strategic, so it may increase its payoff by submitting a false bid (i.e., unequal to its true cost). Here, the payoff of an arm is the difference between the received payment and its true cost. In particular, since the rewards of arms are unknown in advance, it is more difficult for the player to stimulate all strategic arms to report their true cost as the bids (a.k.a. truthfulness [10]). Furthermore, the proposed arm-pulling policy also needs to ensure that each strategic arm has a non-negative payoff (a.k.a. individual rationality [11]). Otherwise, the arms, as rational individuals, will be unwilling to participate in the ACMAB model. In addition, compared with the traditional MAB problem, it is more challenging to analyze the regret bound in the ACMAB problem.

In fact, the proposed ACMAB model can be adopted in many real-world applications. For example, in the user selection problem of mobile crowdsensing [12]–[14], the platform intends to outsource lots of small-scale homogeneous tasks to N registered users under the budget constraint. The user selection process is divided into multiple rounds. In each round, each user can perform only one task and the platform would select K ($< N$) users. The quality of a user completing these tasks is unknown (i.i.d. over time) for the platform a priori. The cost of performing a task for a user is fixed, but it is only known to the user itself. Since the users, as some individual persons, are selfish and rational, they may misreport their true cost to increase their payoff. The platform's objective is to maximize the total achieved quality under the budget constraint. Fundamentally speaking, the user selection problem is equivalent to the proposed ACMAB model. That is, the platform and users are seen as the player and arms, respectively, while the quality of completing tasks is seen as an arm's reward in the ACMAB model.

For the ACMAB problem, the player first needs to face the uncertainty of the arms' rewards. To this end, we adopt the idea of upper confidence bound (UCB), which is always optimistic about the uncertainty, to deal with the tradeoff between exploration and exploitation in the multi-armed bandit problem. More specifically, we carefully design the UCB-based reward expression, i.e., we let the average empirical rewards plus the tailor-made bonus denote the UCB-based reward, in which the bonus is related to the regret bound. Then we divide the ACMAB problem into two sub-problems in each round: winning arm selection problem and payment computation problem. For the winning arm selection problem, we use the ratio of UCB-based reward to the corresponding bid as the selection criteria. At the beginning of each round, we first sort all N arms in the decreasing order of their ratio values, and then select the top K arms according to this order. For the payment computation problem, we compute the critical payment for each winning arm to ensure the truthfulness of the auction mechanism. That is, we first identify the critical arm in a round, i.e., the $(K+1)$-th arm in the completed order of this round. The critical arm means that the $(K+1)$-th arm will win when excluding a winning arm from the arm candidates. Then, we can determine the critical payment based on the detailed computation method. After that, all arms' parameters are updated for the next rounds.

Our major contributions are summarized as follows:

- We combine the auction into the combinatorial multi-armed bandit model (i.e., ACMAB problem), in which each arm is strategic about its true cost. We carefully design a UCB-based reward expression for ACMAB, i.e., the average empirical rewards plus a tailor-made bonus.
- We divide the ACMAB problem into two sub-problems: winning arm selection and payment computation. For the former, we adopt the greedy arm-pulling policy, i.e., selecting the top K arms according to the decreasing order of the ratio of UCB-based rewards to bids; for the latter, we determine the critical payment for each winning arm in a round.
- We devise an auction-based UCB algorithm, called AUCB, to solve the ACMAB problem. We derive an upper bound on regret for AUCB, i.e., $O(NK^3 \ln(B + NK^2 \ln(NK^2)))$, where N means the number of total arms, K denotes the number of arms selected in a round, and B is the budget. Meanwhile, we prove that AUCB can guarantee truthfulness, individual rationality, and computational efficiency.
- We conduct extensive simulations to verify the significant performance of AUCB, and the results show that the total rewards achieved by AUCB are at least 12.49% higher than those of state-of-the-art algorithms.

The remainder of the paper is organized as follows. We first review the related work in Section II. Then, we introduce the ACMAB model and the optimization problem in Section III. Next, we design the AUCB algorithm in Section IV, and analyze the theoretical results of AUCB in Section V. After evaluating the performance of the proposed AUCB algorithm in Section VI, we conclude the paper in Section VII.

II. Related Work

So far, there has been lots of research on the auction mechanism [15]–[17] or CMAB problem [18]–[20]. Nevertheless, few existing works have considered the combination of auction and CMAB problems. On the one hand, researches have focused by far on the auction mechanism design in various fields, such as resource allocation in cloud computing [21]–[23], task assignment in mobile crowdsensing [24]–[28], spectrum sharing in cognitive radio networks [17], [29], and so on [16]. For example, the works [21], [23] study the VM provisioning and allocation problem in cloud computing by using the auction approach. The authors of [27] consider two system models: a platform-centric model and a user-centric model, where both Stackelberg game and auction based incentive mechanisms are proposed. The authors of [24], [25] design online auction mechanisms for dynamic mobile crowdsourcing, while the authors of [17] propose a privacy-preserving auction mechanism, which can improve spectrum allocation efficiency by stimulating users to truthfully reveal their valuations of spectrum.

On the other hand, there are a great number of works that study the CMAB problem [6], [7], [30]–[32]. For example, the authors of [6] consider general CMAB problems and then propose an efficient algorithm which can achieve $O(\ln n)$ regret (n is the number of total rounds). Similarly, the authors of [7] devise an efficient policy for the CMAB problem, which cannot make regret grow logarithmically with total rounds, but does ensure that the required storage would grow linearly in the number of unknown parameters. Also, the authors of [30], [32] study both the stochastic and adversarial CMAB models under a budget constraint for pulling cost. They analyze the upper bound on the regret (i.e., $O(NK^4 \ln B)$) through a rigorous proof. Unfortunately, all of them have neglected the strategic behavior of arms in the CMAB models.

In fact, only a few researches have studied the combination of auction and CMAB problems [4], [33]–[36]. Among them, [4] designs a context-aware MAB incentive mechanism for the quality-based worker selection problem in crowdsensing, where a modified Thompson sampling algorithm is devised. Both works [33], [34] focus on the auction-based MAB problem in the filed of internet advertising, where the click-through rate for advertisements are taken into consideration. The authors of [35] consider a general MAB model with strategic arms and this problem is modeled as an approximate Nash equilibrium among all arms. Nevertheless, all of them differ from our model and problem. The research introduced by [37] is the most related to our work. However, three key differences are listed as follows: 1) the exploration and exploitation phases are separate in [37], i.e., the player will use a fraction of budget to learn and consider the auction design in the remaining budget; 2) each arm has a fixed deadline and only one arm is selected in a round; 3) the method and result of the regret bound is different from ours.

III. Model & Problem

A. Model

Consider that there are N arms, denoted as $\mathcal{N} = \{1, \cdots, i, \cdots, N\}$. In each round t, we use a normalized nonnegative random variable $r_i^t \in [0,1]$ to denote the reward of pulling an arm $i \in \mathcal{N}$. Here, each arm $i \in \mathcal{N}$ is characterized by a random reward distribution and we let r_i denote the expected reward. In other words, the values of $\{r_i^t | i \in \mathcal{N}, \forall t \geq 1\}$ follow an unknown independent and identically distribution with an unknown expectation r_i, i.e., $r_i = \mathbb{E}[r_i^t]$. On the other hand, pulling an arm i will incur a cost and each arm $i \in \mathcal{N}$ has the strategic behavior about its cost. That is to say, the cost claimed by each arm will not always equal to its true cost. Let c_i denote the true cost of the arm i and let b_i denote the cost claimed by the arm i, which is called "bid" in the auction mechanism. Here, we consider that the cost c_i and the bid b_i for each arm $\forall i \in \mathcal{N}$ is located in the range $[c_{min}, c_{max}]$. We use $\mathcal{B} = \{b_i | i \in \mathcal{N}\}$ to denote the bid vector. At the beginning of the auction, each arm will submit their bids to the player. In round t, if the player selects an arm $i \in \mathcal{N}$, he can obtain the reward value r_i^t at the end of this round. Also, the player has to make payment to the winning arms irrespective of the rewards being generated in this round. During this process, the player has to face the strategic cost generated from each arm. Thus, the designed mechanism should ensure the truthfulness (i.e., telling the truth) of arms. In addition, the player has a limited budget, denoted as B. The player wants to maximize the achieved reward under the budget B, while guaranteeing the truthfulness of the strategic arms.

B. Problem Formulation

Actually, the whole auction-based bandit process includes two key problems. The first one is how the player should select K arms in each round so that the expected reward can be maximized under the budget constraint. The second one is how the player should determine the payment for each selected arm in each round, while the truthfulness can be guaranteed.

TABLE I
Description of major notations.

Variable	Description
i, t	the indexes for arm and round, respectively.
N, $\mathcal{N}$	the number of total arms and the set of arms.
K, B	the number of arms selected in a round and budget.
r_i^t	the achieved reward of arm i in round t.
r_i	the mean of the distribution $\{r_i^t \| t \geq 1\}$.
c_i, b_i	the true cost and the claimed bid of arm i.
c_{min}	the minimum values among all costs/bids.
c_{max}	the maximum values among all costs/bids.
Φ^t	the set of arms selected in round t.
p_i^t, $\mathcal{P}^t$	the payment of i and payment vector in round t.
$\overline{r}_i(t)$	the average empirical reward of i until round t.
$\beta_i(t)$	the total times of i being selected until round t.
$u_i(t)$	$u_i(t) = \sqrt{\frac{(K+1)\ln t}{\beta_i(t)}}$ is upper confidence factor.
$\widehat{r}_i(t)$	the UCB-based reward of arm i until round t.
$\alpha_i(t)$	the counter of arm i until round t.

For simplicity, we use Φ^t, where $|\Phi^t| = K$ for $\forall t > 1$, to denote the set of arms selected in round t. Here, $i \in \Phi^t$ means the arm i will be selected by the player in round t, and $i \notin \Phi^t$ otherwise. At each round, the player first needs to select the set of arms so that he can maximize the total rewards and at the same time compute the payment for each winning arm. Here, we use $p_i^t(b_i)$ to denote the corresponding payment for the selected arm i with the bid b_i in round t. When an arm is not selected, the payment is 0. Let $\mathcal{P} = \{\mathcal{P}^1, \cdots, \mathcal{P}^t, \cdots\}$ denote the payment vector, in which $\mathcal{P}^t = \{p_i^t(b_i) | i \in \mathcal{N}\}$ means the payment vector in round t. Based on this, we formalize the winning arm selection problem as follows.

$$\mathit{Maximize}: \quad \mathbb{E}\Big[\sum_t \sum_{i \in \Phi^t} r_i^t\Big] \tag{1}$$

$$\mathit{Subject\ to}: \; \sum_t \sum_{i \in \Phi^t} p_i^t(b_i) \leq B \tag{2}$$

$$|\Phi^t| = K \text{ for } \forall t > 1 \tag{3}$$

Next, we introduce the payment determination problem. The objective is to determine the payment for each winning arm so that the whole auction model satisfies the truthfulness and individual rationality, which are defined as follows:

***Definition** 1:* [*Truthfulness*] [11], [21], [22] For each winning bid b_i, we use $p_i^t(c_i) - c_i$ and $p_i^t(b_i) - c_i$ to denote the i-th arm's payoffs for the truthful and untruthful bids, respectively. The truthful mechanism means that

$$p_i^t(c_i) - c_i \geq p_i^t(b_i) - c_i. \tag{4}$$

The truthfulness of the auction mechanism can guarantee that each strategic arm will report its true cost as the bid, since an untruthful bid will lead to a worse payoff.

***Definition** 2:* [*Individual Rationality*] [26], [38] Since each arm in the ACMAB model is rational, its payoff, defined as the difference between the payment and its true cost, must be greater than or equal to 0; that is, $p_i^t(b_i) - c_i \geq 0$.

For each strategic arm, the payment it received must be greater than or equal to its true cost; otherwise, the arm might be unwilling to participate the ACMAB problem.

***Definition** 3:* [*Computational Efficiency*] [24], [25] An auction-based algorithm has the computational efficiency, indicating that it can be conducted in polynomial time.

Facing the uncertainty of arms, the player's objective is to maximize the total achieved rewards under the budget constraint. Comparing to the case where the expected rewards of arms are known in advance, the proposed policy for the unknown case cannot acquire the optimal rewards.

Definition 4: [*Regret*] [2], [5] Regret means the difference in the total achieved rewards between the optimal policy which knows the expected rewards of arms in advance and the proposed solution for the unknown case, i.e.,

$$R(B) = BR^*/C^* - \sum_{t=1}^{\tau(B)} \sum_{i\in\Phi^t} r_i^t, \quad (5)$$

where $R^* = \sum_{i\in\Phi^*} r_i$ and $C^* = \sum_{i\in\Phi^*} b_i$ mean the total rewards and payments of the optimal solution in a round, respectively, and Φ^* denotes the optimal set of arms for the known case. Meanwhile, $\tau(B)$ represents the total rounds of the proposed algorithm under the budget constraint.

Additionally, we summarize the commonly used notations throughout the paper in Table I.

IV. Algorithm Design

A. Basic Idea

In the Auction-based Combinatorial Multi-Armed Bandit (ACMAB) problem, we need to solve two key sub-problems in each round: the winning arm selection problem and the payment computation problem. For the winning arm selection problem, the player aims to maximize the total achieved rewards under the budget constraint. For the payment computation problem, the player needs to determine the suitable payment for each winning arm so that each strategic arm can tell the truth (i.e., truthfulness). Actually, only when all of the arms are truthful can the player make an efficient arm-pulling policy to maximize the total achieved rewards.

In order to handle the tradeoff between exploitation (i.e., selecting the best arms based on the sampling results) and exploration (i.e., trying some sub-optimal arms to find the potentially optimal arms) in the winning arm selection problem, we first introduce the idea of Upper Confidence Bound (UCB [1], [7], [31]). More specifically, we first use $\beta_i(t)$ to denote the number of the arm $i \in \mathcal{N}$ being selected until the t-th round, and then use $\overline{r}_i(t)$ to denote the average empirical reward of the arm i until round t. The values of $\beta_i(t)$ and $\overline{r}_i(t)$ are updated as follows.

$$\beta_i(t) = \begin{cases} \beta_i(t-1)+1; & i \in \Phi^t \\ \beta_i(t-1); & i \notin \Phi^t \end{cases} \quad (6)$$

$$\overline{r}_i(t) = \begin{cases} \dfrac{\overline{r}_i(t-1)\beta_i(t-1) + r_i^t}{\beta_i(t-1)+1}; & i \in \Phi^t \\ \overline{r}_i(t-1); & i \notin \Phi^t \end{cases} \quad (7)$$

Furthermore, we let $\widehat{r}_i(t)$ denote the UCB-based reward for each arm. The idea of UCB always has the optimism in the face of uncertainty. $\widehat{r}_i(t)$ is described as follows.

$$\widehat{r}_i(t) = \overline{r}_i(t) + u_i(t), \text{ where } u_i(t) = \sqrt{\frac{(K+1)\ln t}{\beta_i(t)}}. \quad (8)$$

Then, we will introduce the winning arm selection and payment computation procedures in detail. Note that we use Φ^t and $\mathcal{P}^t$ to denote the set of selected arms and the corresponding payment in round t. The first is the initialization phase (i.e., $t = 1$), where the player will select all arms (i.e., $\Phi^1 = \mathcal{N}$) to initialize the parameters $\beta_i(t)$, $\overline{r}_i(t)$, and $\widehat{r}_i(t)$. Here, the player uses the value of c_{max} to denote the payment for each arm, in which all values of costs and bids are located in the range $[c_{min}, c_{max}]$. In such a case, each arm's payment received from the player is larger than its true cost, so the payoff for each arm is non-negative (i.e., individual rationality). Next, the player will update the remaining budget.

After the initialization phase, the player will always select K arms in each round under the budget constraint. More specifically, at the beginning of each round t, the player can acquire the values of $\widehat{r}_i(t-1)$ for $\forall i \in \mathcal{N}$, and will initialize Φ^t and $\mathcal{P}^t$. Then, the player will sort the N arms in the decreasing order of $\frac{\widehat{r}_i(t-1)}{b_i}$, in which $\frac{\widehat{r}_i(t-1)}{b_i}$ means the ratio of the UCB-based reward to bid. According to this order, the player will select the top K arms and add them into Φ^t. After that, the player begins to calculate the corresponding payment $\mathcal{P}^t$. Here, we determine the critical payment for each selected arm in a round. The specific payment computation method is denoted as follows:

$$p_i^t(b_i) = \min\{\tfrac{\widehat{r}_i(t-1)}{\widehat{r}_{K+1}(t-1)} \cdot b_{K+1}, c_{max}\}. \quad (9)$$

Here, $\frac{\widehat{r}_i(t-1)}{\widehat{r}_{K+1}(t-1)} b_{K+1}$ means the critical payment and $\min\{\cdot\}$ ensures that the payment will not exceed the maximum value. The critical payment indicates that a winning arm that claims a bid larger than the critical payment will not win in the auction process. However, a smaller bid will always win.

After computing the payment for each winning arm, the player determines the total payment in this round. If the total payment in this round is larger than the remaining budget, the process will terminate. Else, the player selects the arms in Φ^t, observes their rewards in this round, and then updates the parameters of $\beta_i(t)$, $\overline{r}_i(t)$, $\widehat{r}_i(t)$, and the remaining budget. The arm selection and payment computation process will continue until the budget exhausts.

B. Detailed Algorithm

According to the above solution, we devise the auction-based UCB algorithm (called AUCB), as shown in Alg. 1. First, in the initialization phase, i.e., Steps 1-4, the player first selects all of the arms in the first round (i.e., $\Phi^1 = \mathcal{N}$) and obtains the reward values in this round. After that, the player determines the payment for each winning arm, i.e., $p_i^t = c_{max}$. In such a way that each arm's payoff is larger than 0, so the process can ensure the individual rationality. Next, the player can update several parameters such as $\beta_i(t)$, $\overline{r}_i(t)$, $\widehat{r}_i(t)$, $B(t) = B(t-1) - N \cdot c_{max}$, and $r(B) = r(B) + \sum_{i\in\Phi^t} r_i^t$, in which $B(t)$ means the remaining budget after round t.

After the initialization phase, the player begins to select K arms and calculates the payment for each winning arm in each round. In Step 6, the player first updates the index for the round (i.e., $t \Leftarrow t+1$), and then initializes the parameters Φ^t and $\mathcal{P}^t$. In Steps 7-8, the player sorts the N arms in the decreasing order of $\frac{\widehat{r}_i(t-1)}{b_i}$. In Step 9, the player selects the top K arms according to this order, denoted as Φ^t. In Step 10, the player computes the payment for each winning arm based on the definition of critical payment in Eq.(9). After that, the player determines the total payment in this round. As shown in Steps 11-12, if the total payment in this round exceeds the remaining budget, the process will terminate and output

Algorithm 1 Auction-based UCB Algorithm (AUCB)

```
Require: $\mathcal{N}$, $\mathcal{B}$, $K$, and $B$
Ensure: $\Phi$, $r(B)$, $\tau(B)$, and $\mathcal{P}$
1: Initialization: $t=1$, $B(0)=B$, and $r(B)=0$, the player selects all arms in the first round, i.e., $\Phi^1=\mathcal{N}$;
2: Obtain the reward values $r_i^1$ for $\forall i\in\mathcal{N}$ in the first round;
3: Determine the payments for selected arms, i.e., $p_i^1=c_{max}$;
4: Update the parameters: $\overline{r}_i(t)$, $\widehat{r}_i(t)$, $B(t)=B(t-1)-N\cdot c_{max}$, and $r(B)=r(B)+\sum_{i\in\Phi^t} r_i^t$;
5: while $true$ do
6:     $t \Leftarrow t+1$, $\Phi^t=\phi$, and $p_i^t(b_i)=0$ for $\forall i\in\mathcal{N}$;
7:     Sort the arms according to the value $\frac{\widehat{r}_i(t-1)}{b_i}$;
8:     Consider $\frac{\widehat{r}_{i_1}(t-1)}{b_{i_1}} \geq \cdots \geq \frac{\widehat{r}_{i_j}(t-1)}{b_{i_j}} \cdots \geq \frac{\widehat{r}_{i_N}(t-1)}{b_{i_N}}$;
9:     Select the top $K$ arms, denoted as $\Phi^t$;
10:    Compute the payments for each selected arm in $\Phi^t$, i.e., $p_{i_j}^t(b_{i_j})=\min\{\frac{\widehat{r}_{i_j}(t-1)}{\widehat{r}_{i_{K+1}}(t-1)}\cdot b_{i_{K+1}}, c_{max}\}$;
11:    if $\sum_{i\in\Phi^t} p_i^t(b_i) \geq B(t-1)$ then
12:        return Terminate and output $\Phi$, $r(B)$, $\tau(B)=t$, $\mathcal{P}$;
13:    Obtain the rewards $r_i^t$ for $\forall i \in \Phi^t$;
14:    Update the parameters: $\overline{r}_i(t)$, $\widehat{r}_i(t)$, $B(t)=B(t-1)-\sum_{i\in\Phi^t} p_i^t(b_i)$, and $r(B)=r(B)+\sum_{i\in\Phi^t} r_i^t$;
```

the results. Else, the player will update several parameters in Steps 13-14.

V. Performance Analysis

In this section, we analyze the regret bound, truthfulness, individual rationality, and computational efficiency of AUCB.

A. Upper Bound on Regret

First, we analyze the regret bound, which means the total achieved reward gap between the optimal arm-pulling policy and ours (i.e., Definition 4). The optimal policy includes both the optimal winning bid selection solution and optimal payment computation solution according to the known expected rewards of all arms in advance. Here, we use the ratio of the expected rewards to bids, i.e., $\frac{r_i}{b_i}$, as the selection criteria. Meanwhile let the bid (true cost) as the extremely-critical payment, i.e., $p_i = b_i$, for each winning arm. Note that Φ^* and Φ^t mean the optimal set of K strategic arms (the expected rewards are known in advance) and the set of K arms selected in round t, respectively. For the computation of Φ^*, we consider $\frac{r_1}{b_1} \geq \cdots \geq \frac{r_K}{b_K} \geq \frac{r_{K+1}}{b_{K+1}} \geq \cdots \geq \frac{r_N}{b_N}$ and further determine $\Phi^* = \{1, \cdots, K\}$. In the process of analyzing the regret bound, we consider that each arm $i\in\mathcal{N}$ is truthful, i.e., $b_i = c_i$. In the next section, we will prove the truthfulness of AUCB to validate the presupposition. For simplicity of following descriptions, we let R^* and C^* denote the total expected rewards and the total expected payments of the optimal set, respectively. That is to say, we have

$$R^* = \textstyle\sum_{i\in\Phi^*} r_i,\;\; C^*=\sum_{i\in\Phi^*} c_i. \tag{10}$$

Note that in this paper, we always use * to denote the corresponding identifications of the optimal set of arms. Then, we define the smallest and largest possible difference of the ratios of rewards to bids (also the total rewards) among all non-optimal sets of arms $\Phi^t\neq\Phi^*$:

$$\triangle_{max} = \textstyle\sum_{i\in\Phi^*} \frac{r_i}{b_i} - \min_{\Phi^t\neq\Phi^*} \sum_{i\in\Phi^t} \frac{r_i}{b_i}, \tag{11}$$

$$\triangle_{min} = \textstyle\sum_{i\in\Phi^*} \frac{r_i}{b_i} - \max_{\Phi^t\neq\Phi^*} \sum_{i\in\Phi^t} \frac{r_i}{b_i}; \tag{12}$$

$$\nabla_{max} = \textstyle\sum_{i\in\Phi^*} r_i - \min_{\Phi^t\neq\Phi^*} \sum_{i\in\Phi^t} r_i. \tag{13}$$

Moreover, we use a notation $\alpha_i(t)$ to denote the counter for the arm i after the initialization (i.e., $t>1$), in which the counter $\alpha_i(t)$ is updated as follows. In each round $t>1$, when $\Phi^t\neq\Phi^*$, we update the vector $\alpha_i(t)$:

$$i = \text{argmin}_{j\in\Phi^t} \alpha_j(t), \quad \alpha_i(t) = \alpha_i(t) + 1. \tag{14}$$

Here, if multiple arms satisfy the condition, we select any one arm randomly. When the set of arms selected in a round is not the optimal set, one element of the vector $\alpha_i(t)$ will increase by one. This means that the sum of the counter $\alpha_i(t)$ for $\forall i \in \mathcal{N}$ equals to the total number of the sub-optimal sets of arms. Next, we will focus on the upper bound of the counter $\alpha_i(\tau(B))$ where $\tau(B)$ means the largest total rounds of the AUCB algorithm under the budget constraint B. More specifically, we have the following lemma.

Lemma 1: The expected counter $\alpha_i(\tau(B))$ has an upper bound for any arm $i\in\mathcal{N}$, that is,

$$\mathbb{E}[\alpha_i(\tau(B))] \leq \frac{4K^2(K+1)}{(c_{min}\triangle_{min})^2}\ln(\tau(B))+1+\frac{K\pi^2}{3}. \tag{15}$$

Proof: In each round t, one of the following cases must happen: ①the optimal set of arms, i.e., Φ^*, might be selected; ② the player will choose a non-optimal set, i.e., $\Phi^t\neq\Phi^*$. In the former case, the counter $\alpha_i(t)$ will not change, while in the latter case, the counter $\alpha_i(t)$ will be updated according to Eq.(14). Here, we first use the notation $I_i^t\in\{0,1\}$ to denote the indictor, in which $I_i^t=1$ means the corresponding counter $\alpha_i(t)$ is incremented in round t, and $I_i^t=0$ otherwise. Based on this, we have the following results:

$$\begin{aligned}
\alpha_i(\tau) &= \sum_{t=2}^{\tau} \mathbb{I}\{I_i^t=1\} = \lambda + \sum_{t=2}^{\tau} \mathbb{I}\{I_i^t=1, \alpha_i(t)\geq\lambda\} \\
&\leq \lambda + \sum_{t=1}^{\tau} \mathbb{I}\{\sum_{j\in\Phi^t} \frac{\widehat{r}_j(t+1)}{b_j} \geq \sum_{j\in\Phi^*} \frac{\widehat{r}_j(t+1)}{b_j}, \alpha_i(t)\geq\lambda\} \\
&\leq \lambda + \sum_{t=1}^{\tau} \mathbb{I}\Big\{ \max_{\lambda\leq\beta_{i_1}(t)\leq\beta_{i_2}(t)\cdots\leq\beta_{i_K}(t)\leq t} \sum_{j=1}^{K} \frac{\widehat{r}_{i_j}(t)}{b_{i_j}} \\
&\qquad \geq \min_{1\leq\beta_{i_1^*}(t)\leq\beta_{i_2^*}(t)\cdots\leq\beta_{i_K^*}(t)\leq t} \sum_{j=1}^{K} \frac{\widehat{r}_{i_j^*}(t)}{b_{i_j^*}} \Big\} \\
&\leq \lambda + \sum_{t=1}^{\tau} \sum_{\beta_1(t)=\lambda}^{t} \cdots \sum_{\beta_K(t)=\lambda}^{t} \sum_{\beta_{i_1^*}(t)=1}^{t} \cdots \sum_{\beta_{i_K^*}(t)=1}^{t} \\
&\qquad \mathbb{I}\Big\{ \sum_{j=1}^{K} \frac{\widehat{r}_{i_j}(t)}{b_{i_j}} \geq \sum_{j=1}^{K} \frac{\widehat{r}_{i_j^*}(t)}{b_{i_j^*}} \Big\},
\end{aligned} \tag{16}$$

in which $\beta_i(t)$ means the total number of the arm i being selected until the round t. According to the definitions of $\alpha_i(t)$ and $\beta_i(t)$, we get $\beta_i(t)\geq\alpha_i(t)$ for $\forall i\in\mathcal{N}$ and $\forall t\geq 1$.

Next, we focus on the bound of $\sum_{j=1}^{K} \frac{\widehat{r}_{i_j}(t)}{b_{i_j}} \geq \sum_{j=1}^{K} \frac{\widehat{r}_{i_j^*}(t)}{b_{i_j^*}}$. The proof follows the ideas provided in the existing work [1], [30], [32]. More specifically, for the following event

$$\textstyle\sum_{j=1}^{K} \frac{\overline{r}_{i_j}(t)+u_{i_j}(t)}{b_{i_j}} \geq \sum_{j=1}^{K} \frac{\overline{r}_{i_j^*}(t)+u_{i_j^*}(t)}{b_{i_j^*}}, \tag{17}$$

we can get that at least one of the following cases must be true (which is based on the proof by contradiction):

$$\sum_{j=1}^{K} \frac{\overline{r}_{i_j^*}(t)}{b_{i_j^*}} \leq \sum_{j=1}^{K} \frac{r_{i_j^*} - u_{i_j^*}(t)}{b_{i_j^*}}; \tag{18}$$

$$\sum_{j=1}^{K} \frac{\overline{r}_{i_j}(t)}{b_{i_j}} \geq \sum_{j=1}^{K} \frac{r_{i_j} + u_{i_j}(t)}{b_{i_j}}; \tag{19}$$

$$\sum_{j=1}^{K} \frac{r_{i_j^*}}{b_{i_j^*}} < \sum_{j=1}^{K} \frac{r_{i_j} + 2 \cdot u_{i_j}(t)}{b_{i_j}} \tag{20}$$

Next, we prove the upper bound of the probability of Eq.(18) and Eq.(19) as follows:

$$\mathbb{P}\Big\{ \sum_{j=1}^{K} \frac{\overline{r}_{i_j^*}(t)}{b_{i_j^*}} \leq \sum_{j=1}^{K} \frac{r_{i_j^*} - u_{i_j^*}(t)}{b_{i_j^*}} \Big\} \leq \sum_{j=1}^{K} \mathbb{P}\Big\{ \overline{r}_{i_j^*}(t) \leq r_{i_j^*} - u_{i_j^*}(t) \Big\}, \tag{21}$$

and

$$\mathbb{P}\Big\{ \sum_{j=1}^{K} \frac{\overline{r}_{i_j}(t)}{b_{i_j}} \geq \sum_{j=1}^{K} \frac{r_{i_j} + u_{i_j}(t)}{b_{i_j}} \Big\} \leq \sum_{j=1}^{K} \mathbb{P}\Big\{ \overline{r}_{i_j}(t) \geq r_{i_j} + u_{i_j}(t) \Big\}. \tag{22}$$

Then, we focus on the bound of $\mathbb{P}\{\overline{r}_{i_j^*}(t) \leq r_{i_j^*} - u_{i_j^*}(t)\}$ and $\mathbb{P}\{\overline{r}_{i_j}(t) \geq r_{i_j} + u_{i_j}(t)\}$. Here, we will apply the Chernoff-Hoeffding bound [1], [39] to analyze the bound.

$$\mathbb{P}\Big\{ \overline{r}_{i_j^*}(t) \leq r_{i_j^*} - u_{i_j^*}(t) \Big\} \leq e^{-2\beta_{i_j^*}(t) u_{i_j^*}(t)^2} = t^{-2(K+1)} \tag{23}$$

Then, we continue Eq.(21) and get the following results:

$$\sum_{j=1}^{K} \mathbb{P}\Big\{ \overline{r}_{i_j^*}(t) \leq r_{i_j^*} - u_{i_j^*}(t) \Big\} \leq K \cdot t^{-2(K+1)} \tag{24}$$

At the same time, we can also prove that Eq.(22) has the same upper bound. Next, we will choose a certain value λ to make the event $\sum_{j=1}^{K} \frac{r_{i_j^*}}{b_{i_j^*}} < \sum_{j=1}^{K} \frac{r_{i_j} + 2 \cdot u_{i_j}(t)}{b_{i_j}}$ impossible. Based on the fact that $\beta_i(t) \geq \alpha_i(t) \geq \lambda$, we have

$$\begin{aligned} &\sum_{j=1}^{K} \frac{r_{i_j^*}}{b_{i_j^*}} - \sum_{j=1}^{K} \frac{r_{i_j}}{b_{i_j}} - 2 \cdot \sum_{j=1}^{K} \frac{u_{i_j}(t)}{b_{i_j}} \\ &= \sum_{j=1}^{K} \frac{r_{i_j^*}}{b_{i_j^*}} - \sum_{j=1}^{K} \frac{r_{i_j}}{b_{i_j}} - 2 \cdot \sum_{j=1}^{K} \frac{\sqrt{\frac{(K+1)\ln t}{\beta_{i_j}(t)}}}{b_{i_j}} \\ &\geq \triangle_{min} - 2 \cdot \sum_{j=1}^{K} \frac{\sqrt{\frac{(K+1)\ln t}{\beta_{i_j}(t)}}}{b_{i_j}} \\ &\geq \triangle_{min} - 2 \cdot \sum_{j=1}^{K} \frac{\sqrt{\frac{(K+1)\ln t}{\lambda}}}{c_{min}} \geq 0 \end{aligned} \tag{25}$$

After analyzing Eq.(25), we can conclude that Eq.(25) will always hold if λ satisfies the following condition:

$$\lambda \geq \frac{4K^2(K+1)\ln \tau(B)}{(c_{min} \cdot \triangle_{min})^2}. \tag{26}$$

Next, we continue Eq.(16) and get

$$\begin{aligned} \mathbb{E}[\alpha_i(\tau)] &\leq \left\lceil \frac{4K^2(K+1)\ln \tau(B)}{(c_{min} \cdot \triangle_{min})^2} \right\rceil \\ &\quad + \sum_{t=1}^{+\infty} (t - \lambda + 1)^K t^K 2K t^{-2(K+1)} \\ &\leq \frac{4K^2(K+1)\ln \tau(B)}{(c_{min} \cdot \triangle_{min})^2} + 1 + 2K \sum_{t=1}^{+\infty} t^{-2} \\ &\leq \frac{4K^2(K+1)\ln \tau(B)}{(c_{min} \cdot \triangle_{min})^2} + 1 + \frac{K\pi^2}{3} = \varphi_1 \ln \tau(B) + \varphi_2, \end{aligned} \tag{27}$$

where $\begin{cases} \varphi_1 = \dfrac{4(K+1)K^2}{(\triangle_{min} c_{min})^2}; \\ \varphi_2 = 1 + \dfrac{K\pi^2}{3}. \end{cases}$ Hence, the lemma holds. ■

Since the bound of $\alpha_i(\tau(B))$ is highly related to the expected number of total rounds in the AUCB algorithm, we then analyze the upper bound of $\tau(B)$. $\tau(B)$ represents the total rounds of the AUCB algorithm under the budget B. Here, we also define another payment computation method in the case where the expected rewards are known in advance. That is, according to the selection criteria $\frac{r_i}{b_i}$, we let $p_i(b_i) = \frac{r_i \cdot b_{K+1}}{r_{K+1}} \geq b_i$ be the payment for each winning arm i. Thus, the total payment in each round, denoted as $C^\star$, is determined and we have

$$C^\star = \sum_{i \in \Phi^*} \frac{r_i \cdot b_{K+1}}{r_{K+1}} \geq \sum_{i \in \Phi^*} b_i = C^*. \tag{28}$$

For the stopping round $\tau(B)$, we have the following lemma.

***Lemma** 2:* The stopping round of the AUCB algorithm under the budget B, i.e., $\tau(B)$, is bounded as follows:

$$\frac{B}{C^\star} - \varphi_1 \varphi_4 \ln\left(\frac{2B}{C^*} + \varphi_3\right) - \varphi_2 \varphi_4 - 1 \leq \tau(B) \leq \frac{2B}{C^*} + \varphi_3, \tag{29}$$

in which $\begin{cases} \varphi_3 = \dfrac{2Nc_{max}}{Kc_{min}} \left(\varphi_2 - \varphi_1 + \varphi_1 \ln\left(\dfrac{2Nc_{max}\varphi_1}{Kc_{min}}\right)\right); \\ \varphi_4 = \dfrac{Nc_{max}}{C^\star}. \end{cases}$

Proof: We first let $\tau^*(B)$ denote the stopping round of the optimal solution under the budget B. Since the optimal solution knows the expected rewards of all arms in advance and the bids submitted by each strategic arm are fixed, the optimal set of arms selected in each round is determined, i.e., Φ^*. Then, the payment for each winning arm can be determined and the total payment in one round can be calculated according to Eq.(10). Thus, the number of total rounds is fixed, i.e., $\tau^*(B) = \lfloor B/C^* \rfloor$, and further we have the following results:

$$B/C^* - 1 \leq \tau^*(B) \leq B/C^*. \tag{30}$$

When deriving the upper bound of $\tau(B)$, we first analyze the relationship between $\tau^*(B)$ and $\tau(B)$. According to the payment computation, we get that the payment for each winning arm in AUCB algorithm is greater than its true cost. Thus, the total payment in each round is greater than the value $K \cdot c_{min}$, and we have $\tau(B) \leq \frac{B}{Kc_{min}}$. Then, we have

$$\begin{aligned} \tau(B) &\leq \tau^*(B) + \tau\left(\sum_{i \notin \Phi^*} \beta_i(\tau(B)) \cdot c_{max}\right) \\ &\leq \tau^*(B) + c_{max} \cdot \tau\left(\sum_{i=1}^{N} \alpha_i(\tau(B))\right) \\ &\leq \tau^*(B) + \frac{N \cdot c_{max}}{K \cdot c_{min}} \mathbb{E}[\alpha_i(\tau(B))] \\ &\leq \frac{B}{C^*} + \frac{N \cdot c_{max}}{K \cdot c_{min}} (\varphi_1 \ln(\tau(B)) + \varphi_2). \end{aligned} \tag{31}$$

According to the inequality $\ln x \leq x - 1$ for $\forall x > 0$, we have the following results:

$$\begin{aligned} &\ln\left(\frac{Kc_{min}}{2Nc_{max}\varphi_1} \tau(B)\right) \leq \frac{Kc_{min}}{2Nc_{max}\varphi_1} \tau(B) - 1 \\ &\Rightarrow \ln(\tau(B)) \leq \frac{Kc_{min}}{2Nc_{max}\varphi_1} \tau(B) - 1 + \ln\left(\frac{2Nc_{max}\varphi_1}{Kc_{min}}\right) \end{aligned} \tag{32}$$

By substituting the bound of $\ln(\tau(B))$ in Eq.(32) into Eq.(31), we have

$$\tau(B) \leq \frac{B}{C^*} + \frac{N \cdot c_{max}}{K \cdot c_{min}} (\varphi_1 \ln(\tau(B)) + \varphi_2)$$

$$
\begin{aligned}
&\leq \tfrac{B}{C^*} + \tfrac{1}{2}\tau(B) + \tfrac{Nc_{max}}{Kc_{min}}(\varphi_2 - \varphi_1 + \varphi_1 \ln(\tfrac{2Nc_{max}\varphi_1}{Kc_{min}}))\\
&\Rightarrow \tau(B) \leq \tfrac{2B}{C^*} + \tfrac{2Nc_{max}}{Kc_{min}}(\varphi_2 - \varphi_1 + \varphi_1 \ln(\tfrac{2Nc_{max}\varphi_1}{Kc_{min}})) \qquad (33)
\end{aligned}
$$

For simplicity of following description, we let $\varphi_3 = \frac{2Nc_{max}}{Kc_{min}}(\varphi_2 - \varphi_1 + \varphi_1 \ln(\frac{2Nc_{max}\varphi_1}{Kc_{min}}))$ and have $\tau(B) \leq \frac{2B}{C^*} + \varphi_3$.

Next, we focus on the lower bound of $\tau(B)$. Here, we divide B into two parts: B^* and B^-, in which B^* means the budget is used to select the optimal set of arms and B^- indicates the remaining budget spent on pulling the non-optimal sets of arms. Then, we use $\tau^\star(B)$ to denote the total rounds in which the budget B is given and the payment is calculated by $p_i(b_i) = \frac{r_i \cdot b_{K+1}}{r_{K+1}}$. Hence, we get $\tau^\star(B) \leq \tau^*(B)$ and $B/C^\star - 1 \leq \tau^\star(B) \leq B/C^\star$. Since $\tau^\star(B)$ and $\tau(B)$ are based on the same payment computation method, we have

$$
\begin{aligned}
\tau(B) &= \tau(B^* + B^-) \geq \tau^\star(B^*)\\
&\geq \tau^\star(B - \textstyle\sum_{i \notin \Phi^*} \beta_i(\tau(B)) \cdot c_{max})\\
&\geq \tau^\star(B - c_{max} \cdot \textstyle\sum_{i=1}^{N} \alpha_i(\tau(B)))\\
&\geq \tau^\star(B - c_{max} \cdot N \cdot (\varphi_1 \ln(\tau(B)) + \varphi_2))\\
&\geq \tfrac{B - Nc_{max}(\varphi_1 \ln(\tau(B)) + \varphi_2)}{C^\star} - 1 \qquad (34)
\end{aligned}
$$

We first take the logarithm of Eq.(33) and then substitute the result into Eq.(34). Now, we get the lower bound of $\tau(B)$:

$$
\begin{aligned}
\tau(B) &\geq \tfrac{B}{C^\star} - \tfrac{Nc_{max}(\varphi_1 \ln(2B/C^* + \varphi_3) + \varphi_2)}{C^\star} - 1\\
&= \tfrac{B}{C^\star} - \varphi_1\varphi_4 \ln(\tfrac{2B}{C^*} + \varphi_3) - \varphi_2\varphi_4 - 1, \qquad (35)
\end{aligned}
$$

where we let $\varphi_4 = \frac{Nc_{max}}{C^\star}$ for simplicity.

The lemma holds. ■

At last, we analyze the expected regret of the AUCB algorithm. Here, we let $R(B)$ denote the expected regret, i.e.,

$$R(B) = BR^*/C^* - \sum_{t=1}^{\tau(B)} \sum_{i \in \Phi^t} r_i^t. \qquad (36)$$

Then, we prove the upper bound of the expected regret for our proposed algorithm, and we have the following theorem.

***Theorem** 1:* The expected regret of the AUCB algorithm is bounded as $O\Big(NK^3 \ln(B + NK^2 \ln(NK^2))\Big)$. More specifically, we have the following inequality, where φ_1, φ_2, φ_3, and φ_4 are some fixed constants shown before.

$$
\begin{aligned}
R(B) \leq\ & (\varphi_1\varphi_4 R^* + N\nabla_{max}\varphi_1) \ln(\tfrac{2B}{C^*} + \varphi_3)\\
& + \varphi_2\varphi_4 R^* + N\nabla_{max}\varphi_2 + R^*. \qquad (37)
\end{aligned}
$$

Proof: According to the definition of regret, Lemma 1 and Lemma 2, we have the following results:

$$
\begin{aligned}
R(B) &= \tfrac{B}{C^*}R^* - \tau(B)R^* + \tau(B)R^* - \mathbb{E}[\textstyle\sum_{t=1}^{\tau(B)} \sum_{i \in \Phi^t} r_i^t]\\
&\leq \tfrac{B}{C^*}R^* - \tau(B)R^* + \textstyle\sum_{i=1}^{N} \alpha_i(\tau(B))\nabla_{max}\\
&\leq \tfrac{B}{C^*}R^* - \tau(B)R^* + N\nabla_{max}\Big(\varphi_1 \ln(\tfrac{2B}{C^*} + \varphi_3) + \varphi_2\Big)\\
&\leq \tfrac{B}{C^*}R^* - (\tfrac{B}{C^\star} - \varphi_1\varphi_4 \ln(\tfrac{2B}{C^*} + \varphi_3) - \varphi_2\varphi_4 - 1)R^*\\
&\quad + N\nabla_{max}\Big(\varphi_1 \ln(\tfrac{2B}{C^*} + \varphi_3) + \varphi_2\Big)\\
&\leq (\varphi_1\varphi_4 R^* + N\nabla_{max}\varphi_1) \ln(\tfrac{2B}{C^*} + \varphi_3)\\
&\quad + \varphi_2\varphi_4 R^* + N\nabla_{max}\varphi_2 + R^*. \qquad (38)
\end{aligned}
$$

Thus, the theorem holds. ■

TABLE II
SIMULATION SETTINGS

parameter name	range
budget, B	10^4 – 10^6 (5×10^5 in default)
number of arms, N	50 – 100 (60 in default)
number of selected arms, K	10 – 50 (20 in default)
expected reward, r_i	0.1 – 1
variance of reward, σ_i	0 – $\min\{r_i/3, (1-r_i)/3\}$
cost, c_i and bid, b_i	0.1 – 1

B. Truthfulness

Then, we prove the truthfulness of AUCB in each round according to Definition 1 in the following theorem.

***Theorem** 2:* In each round, AUCB has the truthfulness.

Proof: By considering the Myerson's theorem [10], we know that the auction mechanism is truthful if and only if it satisfies two conditions: 1) the winning arm selection process is monotonic; 2) each winning arm is paid the critical value. First, we can easily prove that our winning arm selection is monotonic in each round. For each bid value b_i, if the arm i can win the auction with b_i in round t, the arm i must win when it submits a smaller value $\widetilde{b}_i = b_i - \theta$ in which $\theta \geq 0$. This conclusion is based on the greedy selection criteria $\frac{\widehat{r}_i(t)}{b_i}$.

Next, we prove that our proposed algorithm will also meet the second condition. For the bid b_i, we remove the arm i and let $\mathcal{N}_{-i}$ denote the new arm set. Then, we re-select the new K arms based on $\mathcal{N}_{-i}$, denoted as Φ_{-i}^t. Due to the elimination of b_i, there must be another arm/bid, denoted as $b_{i'}$, which will be selected to add into the solution. Additionally, due to the selection criteria $\frac{\widehat{r}_{i_1}(t)}{b_{i_1}} \geq \cdots \geq \frac{\widehat{r}_{i_K}(t)}{b_{i_K}} \geq \frac{\widehat{r}_{i_{K+1}}(t)}{b_{i_{K+1}}} \geq \cdots \geq \frac{\widehat{r}_{i_N}(t)}{b_{i_N}}$, the candidate bid for any winning bid is the same, i.e., $b_{i_{K+1}}$. In other words, given all other arms' selection strategies, the critical payment for winning bid b_i is determined by $p_i^t(b_i) = \frac{\widehat{r}_i(t)}{\widehat{r}_{K+1}(t)} b_{K+1}$. In such a case, if the bid claimed by the arm i is lower than $p_i^t(b_i)$, i.e., $b \leq p_i^t(b_i)$, then $\frac{\widehat{r}_i(t)}{b}$ must be greater than $\frac{\widehat{r}_{K+1}(t)}{b_{K+1}}$. This means that the bid b must be selected prior to b_{K+1} according to the selection criteria $\frac{\widehat{r}_i(t)}{b_i}$. However, the payment for the winning bid b is still the same, that is, $p_i^t(b_i) = \frac{\widehat{r}_i(t)}{\widehat{r}_{K+1}(t)} b_{K+1}$. On the other hand, if $b > p_i^t(b_i)$ happens, then the bid b_{K+1} must be selected prior to b. This indicates that the bid b will lose the auction.

By combining the two aspects, we conclude that $p_i^t(b_i) = \frac{\widehat{r}_i(t)}{\widehat{r}_{K+1}(t)} b_{K+1}$ is the critical payment exactly. Therefore, in each round, the AUCB algorithm can ensure the truthfulness of all strategic arms according to the Myerson' theorem [10]. This is because a higher bid (i.e., $b_i > c_i$) will not increase its payment according to $p_i^t(b_i) = \frac{\widehat{r}_i(t)}{\widehat{r}_{K+1}(t)} b_{K+1}$. Furthermore, this action might incur the failure of this arm in the auction process. Thus, the theorem holds ■

C. Individual Rationality

The individual rationality indicates that each arm's payoff is greater than 0 (i.e., Definition 2).

***Theorem** 3:* In each round, AUCB has individual rationality.

Proof: For each arm i, if the bid b_i does not win the auction in round t, the corresponding payoff is 0; otherwise, the payoff is denoted as $p_i^t(b_i) - c_i$. Here, we have $p_i^t(b_i) = \min\{\frac{\widehat{r}_i(t)}{\widehat{r}_{K+1}(t)} \cdot$

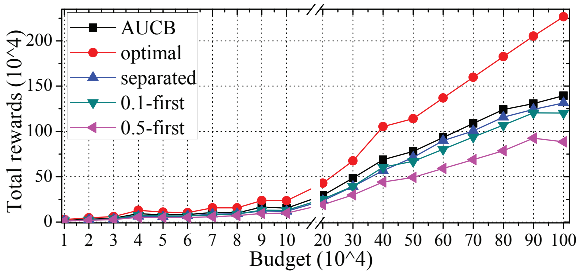

Fig. 1. Total achieved rewards vs. budget

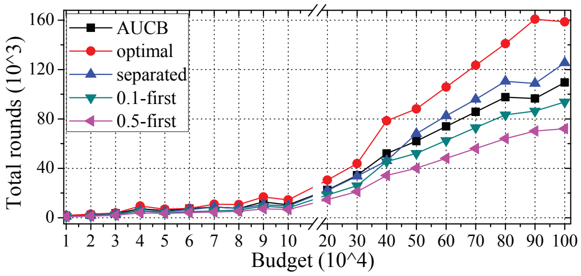

Fig. 2. Total rounds vs. budget

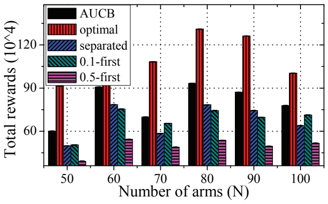

Fig. 3. Rewards vs. N

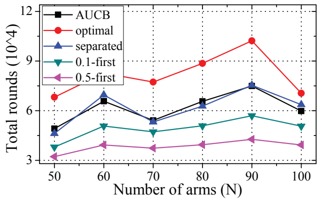

Fig. 4. Rounds vs. N

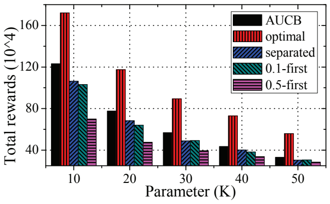

Fig. 5. Rewards vs. K

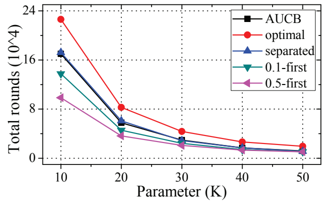

Fig. 6. Rounds vs. K

$b_{K+1}, c_{max}\}$ in the t-th round. Since the arm b_i is selected prior to b_{K+1} in round t, we get $\frac{\widehat{r}_i(t)}{b_i} \geq \frac{\widehat{r}_{K+1}(t)}{b_{K+1}}$. Then, we have $b_i \leq \frac{\widehat{r}_i(t)}{\widehat{r}_{K+1}(t)} \cdot b_{K+1} = p_i^t(b_i)$. Further, based on the truthfulness of AUCB in each round, we have $b_i = c_i$ for any arm $i \in \mathcal{N}$. Thus, we get $c_i \leq \frac{\widehat{r}_i(t)}{\widehat{r}_{K+1}(t)} \cdot b_{K+1} = p_i^t(b_i)$, i.e., $p_i^t(b_i) - c_i \geq 0$, in any round t. We can prove the individual rationality of AUCB in each round and the theorem holds. ■

D. Computational Efficiency

In order to prove the computational efficiency of the proposed AUCB algorithm, we need to demonstrate that AUCB can be conducted in polynomial time. According to Definition 3, we get the following theorem.

Theorem 4: AUCB is computationally efficient.

Proof: By analyzing Alg.1, we get that AUCB has total $\tau(B)$ rounds. In each round, the computational overhead is denoted as $O(N)$, where N means the total number of arms. According to Lemma 2, the stopping round $\tau(B)$ is bounded as $\tau(B) \leq 2B/C^* + \varphi_3$, in which $\varphi_3 = \frac{2Nc_{max}}{Kc_{min}}(\varphi_2 - \varphi_1 + \varphi_1 \ln(\frac{2Nc_{max}\varphi_1}{Kc_{min}}))$. Therefore, we can determine that the algorithmic procedure of Alg. 1 is in polynomial time and the corresponding computational overhead is denoted as $O(NB + N^2K^2\ln(NK^2))$. The theorem holds. ■

VI. Experimental Simulations

A. Experimental Methodology

Compared Algorithms: We design three other algorithms for comparison, called "optimal", "separated", and "ϵ-first". "optimal" means that the algorithm knows the expected rewards of all arms in advance, and then it sorts the N arms in the decreasing order of $\frac{r_i}{b_i}$. Then, the optimal algorithm selects the top K arms according to this order. The extremely-critical payment for each winning arm is computed as $p_i(b_i) = b_i$. The "separated" algorithm [37] divides the budget B into two parts: exploration budget and exploitation budget. Here, it lets $B_1 = \frac{(c_{max}N\ln(NB))^{1/3} * B^{2/3}}{2^{1/3}}$ denote the exploration budget. During the exploration phases, the player will select K arms in sequence from all N arms and lets $p_i^t(b_i) = c_{max}$ denote the payment for each selected arm in a round. Before the exploration budget exhausts, this algorithm always updates the average sampling rewards, denoted as $\overline{r}_i$, in each round. In the exploitation phases, the algorithm uses $\widetilde{r}_i = \overline{r}_i + \sqrt{\frac{Nc_{max}\ln(NB)}{2*B_1}}$ to denote the upper confidence bound of arms' rewards, and it lets $\widetilde{r}_i/b_i$ denote the selection criteria. According to this, it always selects the top K arms in each round under the exploitation budget constraint. The payment is calculated by $p_i^t(b_i) = \min\{\frac{\widetilde{r}_i}{\widetilde{r}_{K+1}}b_{K+1}, c_{max}\}$. Note that in the "separated" algorithm, the average sampling reward results in the exploitation phase will not update, indicating that the exploration and exploitation are separated. While in the ϵ-first algorithm [40], the player will use the first $\epsilon * B$ budget to randomly select K winning arms in each round (i.e., exploration). Based on the exploration results, the player can update several parameters of all arms, such as $\beta_i(t)$ and $\overline{r}_i(t)$ for $\forall i \in \mathcal{N}$. In the remaining $(1-\epsilon) * B$ budget, the player will always select the top K arms in the decreasing order of $\frac{\overline{r}_i(t)}{b_i}$ where $\overline{r}_i(t)$ denotes the average empirical reward of the arm i, and calculate the payment for each winning arm by using $p_i^t(b_i) = \min\{\frac{\overline{r}_i(t)}{\overline{r}_{K+1}(t)}b_{K+1}, c_{max}\}$. In the ϵ-first algorithm, we will change ϵ from 0.1 to 0.5.

Simulation Settings: Then, we present the simulation settings in detail. We let the number of arms (i.e., N) be selected from $\{50, 60, 70, 80, 90, 100\}$, and let $N = 60$ in default. We generate the number of arms selected in each round, i.e., K, from $\{10, 20, 30, 40, 50\}$ and let $K = 20$ in default. Also, we change the budget from $[10^4, 10^6]$ and let $B = 5 \times 10^5$ in default. When generating the rewards of each arm, we let the expected reward (i.e., r_i) be selected from the range $[0.1, 1]$. Note that we adopt the Gaussian distribution to generate the rewards of all arms in each round. In order to ensure that the generated reward values are located in $(0, 1]$, we let the variance of Gaussian distribution for the arm i, denoted as σ_i, be selected from the range $(0, \min\{\frac{r_i}{3}, \frac{1-r_i}{3}\}]$. In such settings, the generated reward values in each round (i.e., r_i^t) is located in the range $(0, 1]$ with the probability of at least 99.7%, according to the properties of Gaussian distribution. Moreover, we use the uniform distribution to generate the cost and the claimed bid for each arm, i.e., c_i and b_i. Since we have proven the truthfulness in Theorem 2, we here let

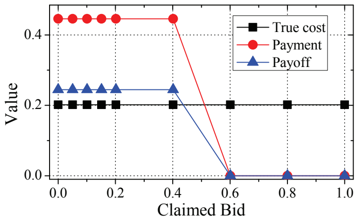

Fig. 7. Truthfulness

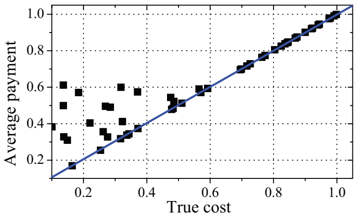

Fig. 8. Individual Rationality

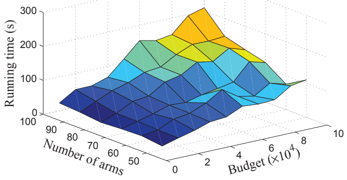

Fig. 9. Computational Efficiency

$b_i = c_i$ for $\forall i \in \mathcal{N}$. We generate c_i randomly from the range $[0.1, 1]$. We adopt two main evaluation metrics: total rewards and total rounds under the budget constraint. In addition, we will also evaluate the truthfulness, individual rationality, and computational efficiency in the simulations. We list the major simulation settings in Table II.

B. Simulation Results

Now, we display and analyze the simulation results in detail. First, when we change the budget from 10^4 to 10^6, we evaluate the achieved total rewards and the total rounds, as shown in Fig. 1 and Fig. 2. We see that the achieved rewards of all four algorithms rise along with the increase of the budget B. Moreover, the performance of AUCB is better than the two compared algorithms, and we calculate that the achieved rewards of AUCB are 19.49% and 21.65% higher than those of the "separated" and 0.1-first algorithms on average, respectively. From Fig. 1, we get that the gap between AUCB and optimal (i.e., "regret") is enlarging with the increase of budget. We also present the relationship between the total rounds and budget in Fig. 2. The variation tendency is similar with Fig. 1. The difference lies in that the total rounds of the "separated" algorithm will be greater than that of AUCB when $B \geq 5 \times 10^5$. This means that the total payment calculated by the "separated" algorithm in a round is smaller than ours. Even so, the total rewards of AUCB are still better than that of "separated". We get that both the total rewards and rounds achieved by AUCB are higher than that of the ϵ-first algorithm.

Furthermore, we analyze the impact of the number of total arms (i.e., N), and display the results in Fig. 3 and Fig. 4. In any settings, the total rewards achieved by AUCB are higher than those of the two compared algorithms. Precisely, the total reward of AUCB is 17.67% and 18.98% larger than those of the "separated" and 0.1-first algorithms on average, respectively. We also evaluate the performance of all algorithms by changing the number of arms selected in each round (i.e., K). We present the simulation results in Fig. 5 and Fig. 6. Along with the increase of the parameter K, the total achieved rewards of all algorithms are going down. This is because the total payment in each round increases, which results in the decrease of total rounds, as shown in Fig. 6. K is used to balance the tradeoff between the achieved rewards and execution time. In such settings, we get that the total rewards achieved by AUCB are about 12.49% and 15.51% higher than those of "separated" and 0.1-first, respectively. These results are consistent with our theoretical analysis.

On the other hand, we evaluate the properties of auction mechanism in the AUCB algorithm. First, we prove the truthfulness of AUCB in each round and present the simulation results in Fig. 7. In any one round, we randomly select one arm and change its bid values. During this process, we ensure that all other settings remain unchanged. From Fig. 7, we see that the true cost is about 0.2 and the critical payment is about 0.45. When the claimed bid is lower than the critical payment, the arm always wins and the corresponding payoff is about 0.25, which does not vary with the change of the bid values. However, when the bid is higher than the critical payment, the arm must fail in the AUCB algorithm. As a result, the payoff becomes 0. Thus, each arm has no incentives to lie in the AUCB algorithm. Next, we analyze the individual rationality of AUCB in Fig. 8. More precisely, we record the payment for each arm in each round and the number of each arm selected under the budget. After the budget is exhausted, we compute the average payment for each arm. We see that each arm's average payment is higher than his true cost. This results prove the individual rationality of AUCB. Finally, we evaluate the computational efficiency of AUCB in Fig. 9. When we change the budget B and the number of arms N, we find that the highest running time is about 200 seconds under the settings of $B = 10^6$ and $N = 100$. These simulation results still remain consistent with our theoretical analysis.

VII. Conclusion

In this paper, we study the budget-constrained auction-based combinatorial multi-armed bandit (ACMAB) problem where each arm is strategic about its cost. In addition to the dilemma between exploration and exploitation, the ACMAB problem also has to face the truthfulness of the strategic cost. Particularly, the player must handle the payment computation problem so that the total rewards can be maximized under the budget constraint. To this end, we adopt the idea of UCB and further propose the AUCB algorithm. AUCB greedily selects the top K arms according to the ratio of UCB-based reward to bid, and meanwhile determines the critical payment for each winning arm in a round. We derive an upper bound on regret for AUCB, i.e., $O(NK^2 \ln(B + NK^2 \ln(NK^2)))$, and prove the truthfulness, individual rationality, and computational efficiency of AUCB. Extensive simulations results show that the total rewards of AUCB are at least 12.49% higher than those of state-of-the-art (e.g., "exploration-separated") algorithms.

Acknowledgments

The work of Guoju Gao, He Huang, and Yu-E Sun was supported by the National Natural Science Foundation of China (NSFC) under Grant 61873177, 62072322, and U20A20182. The work of Mingjun Xiao was supported in part by the NSFC under Grant 61872330, 61936015, and U1709217, in part by Anhui Initiative in Quantum Information Technologies under Grant AHY150300, and in part by the NSF of Jiangsu in China under Grant BK20191194. The work of Jie Wu was supported by NSF Grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, CNS 1651947, and CNS 1564128.

References

[1] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[2] S. Bubeck, N. Cesa-Bianchi *et al.*, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends® in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[3] H. Gupta, A. Eryilmaz, and R. Srikant, "Low-complexity, low-regret link rate selection in rapidly-varying wireless channels," in *IEEE INFOCOM*, 2018.

[4] Y. Wu, F. Li, L. Ma, Y. Xie, T. Li, and Y. Wang, "A context-aware multiarmed bandit incentive mechanism for mobile crowd sensing systems," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7648–7658, 2019.

[5] F. Li, J. Liu, and B. Ji, "Combinatorial sleeping bandits with fairness constraints," in *IEEE INFOCOM*, 2019.

[6] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework and applications," in *International Conference on Machine Learning*, 2013.

[7] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1466–1478, 2012.

[8] L. Anderegg and S. Eidenbenz, "Ad hoc-vcg: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents," in *ACM MobiCom*, 2003.

[9] H. Ma, D. Zhao, and P. Yuan, "Opportunities in mobile crowd sensing," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 29–35, 2014.

[10] R. B. Myerson, "Optimal auction design," *Mathematics of operations research*, vol. 6, no. 1, pp. 58–73, 1981.

[11] G. Gao, M. Xiao, J. Wu, L. Huang, and C. Hu, "Truthful incentive mechanism for nondeterministic crowdsensing with vehicles," *IEEE Transactions on Mobile Computing*, vol. 17, no. 12, pp. 2982–2997, 2018.

[12] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *IEEE INFOCOM*, 2015.

[13] S. Yang, F. Wu, S. Tang, T. Luo, X. Gao, L. Kong, and G. Chen, "Selecting most informative contributors with unknown costs for budgeted crowdsensing," in *IEEE/ACM IWQoS*, 2016.

[14] G. Gao, J. Wu, M. Xiao, and G. Chen, "Combinatorial multi-armed bandit based unknown worker recruitment in heterogeneous crowdsensing," in *IEEE INFOCOM*, 2020.

[15] S. J. Rassenti, V. L. Smith, and R. L. Bulfin, "A combinatorial auction mechanism for airport time slot allocation," *The Bell Journal of Economics*, pp. 402–417, 1982.

[16] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "A double-auction mechanism for mobile data-offloading markets," *IEEE/ACM Transactions on Networking*, vol. 23, no. 5, pp. 1634–1647, 2014.

[17] Q. Huang, Y. Tao, and F. Wu, "Spring: A strategy-proof and privacy preserving spectrum auction mechanism," in *IEEE INFOCOM*, 2013.

[18] S. Tang, Y. Zhou, K. Han, Z. Zhang, J. Yuan, and W. Wu, "Networked stochastic multi-armed bandits with combinatorial strategies," in *IEEE ICDCS*, 2017.

[19] Z. Qin, X. Gan, J. Liu, H. Wu, H. Jin, and L. Fu, "Exploring best arm with top reward-cost ratio in stochastic bandits," in *IEEE INFOCOM*, 2020.

[20] J. Vermorel and M. Mohri, "Multi-armed bandit algorithms and empirical evaluation," in *European conference on machine learning*. Springer, 2005.

[21] M. M. Nejad, L. Mashayekhy, and D. Grosu, "Truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 2, pp. 594–603, 2015.

[22] H. Zhang, H. Jiang, B. Li, F. Liu, A. V. Vasilakos, and J. Liu, "A framework for truthful online auctions in cloud computing with heterogeneous user demands," *IEEE Transactions on Computers*, vol. 65, no. 3, pp. 805–818, 2016.

[23] L. Zhang, Z. Li, and C. Wu, "Dynamic resource provisioning in cloud computing: A randomized auction approach," in *IEEE INFOCOM*, 2014.

[24] D. Zhao, X.-Y. Li, and H. Ma, "Budget-feasible online incentive mechanisms for crowdsourcing tasks truthfully," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 647–661, 2016.

[25] Y. Wei, Y. Zhu, H. Zhu, Q. Zhang, and G. Xue, "Truthful online double auctions for dynamic mobile crowdsourcing," in *IEEE INFOCOM*, 2015.

[26] Y. Chen, B. Li, and Q. Zhang, "Incentivizing crowdsourcing systems with network effects," in *IEEE INFOCOM*, 2016.

[27] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *ACM MobiCom*, 2012.

[28] M. Xiao, J. Wu, L. Huang, R. Cheng, and Y. Wang, "Online task assignment for crowdsensing in predictable mobile social networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2306–2320, 2017.

[29] G. S. Kasbekar and S. Sarkar, "Spectrum auction framework for access allocation in cognitive radio networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 6, pp. 1841–1854, 2010.

[30] Y. Xia, T. Qin, W. Ma, N. Yu, and T.-Y. Liu, "Budgeted multi-armed bandits with multiple plays," in *IJCAI*, 2016.

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

[32] D. P. Zhou and C. J. Tomlin, "Budget-constrained multi-armed bandits with multiple plays," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[33] R. Gonen and E. Pavlov, "An incentive-compatible multi-armed bandit mechanism," in *ACM symposium on Principles of distributed computing*, 2007, pp. 362–363.

[34] M. Babaioff, Y. Sharma, and A. Slivkins, "Characterizing truthful multi-armed bandit mechanisms," *SIAM Journal on Computing*, vol. 43, no. 1, pp. 194–230, 2014.

[35] M. Braverman, J. Mao, J. Schneider, and S. M. Weinberg, "Multi-armed bandit problems with strategic arms," in *Conference on Learning Theory*, 2019, pp. 383–416.

[36] S. Jain, B. Narayanaswamy, and Y. Narahari, "A multiarmed bandit incentive mechanism for crowdsourcing demand response in smart grids," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

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

[38] A. Singla and A. Krause, "Truthful incentives in crowdsourcing tasks using regret minimization mechanisms," in *ACM international conference on World Wide Web*, 2013.

[39] J. P. Schmidt, A. Siegel, and A. Srinivasan, "Chernoff–hoeffding bounds for applications with limited independence," *SIAM Journal on Discrete Mathematics*, vol. 8, no. 2, pp. 223–250, 1995.

[40] L. Tran-Thanh, A. Chapman, E. M. de Cote, A. Rogers, and N. R. Jennings, "Epsilon–first policies for budget–limited multi-armed bandits," in *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.