k-Level Truthful Incentivizing Mechanism and Generalized k-MAB Problem

Pengzhan Zhou, *Student Member, IEEE,* Xin Wei, *Student Member, IEEE,* Cong Wang, *Member, IEEE,* and Yuanyuan Yang, *Fellow, IEEE*

Abstract—Multi-armed bandits problem has been widely utilized in economy-related areas. Incentives are explored in the sharing economy to inspire users for better resource allocation. Previous works build a budget-feasible incentive mechanism to learn users' cost distribution. However, they only consider a special case that all tasks are considered as the same. The general problem asks for finding a solution when the cost for different tasks varies. In this paper, we investigate this problem by considering a system with k levels of difficulty. We present two incentivizing strategies for offline and online implementation, and formally derive the ratio of utility between them in different scenarios. We propose a regret-minimizing mechanism to decide incentives by dynamically adjusting budget assignment and learning from users' cost distributions. We further extend the problem to a more generalized k-MAB problem by removing the contextual information of difficulties. CUE-UCB algorithm is proposed to address the online advertisement problem for multi-platforms. Our experiment demonstrates utility improvement about 7 times and time saving of 54% to meet a utility objective compared to the previous works in sharing economy, and up to 175% increment of utility for online advertising.

Index Terms—Reinforcement learning, multi-armed bandits, incentivizing mechanism, sharing economy, online advertisement

1 Introduction

Recent trends of applying Reinforcement Learning (RL) mechanisms in economy related areas have shed light on better resolutions to these human-involved fields. Economic problems such as sharing economy, incentivizing mechanisms, online advertising, gambling-like problems are highly complicated due to the dynamic and unpredicted nature of the involving humans. The performance of traditionally heuristic algorithms is diminished in face of the varying cases due to human actions. However, the reinforcement learning can explore and exploit the human factors, which automatically provides ongoing solutions while also converges to the best solutions simultaneously via learning the behaviors of the participants dealt with.

The design of incentivizing mechanisms in the sharing economy is a motivating example. The sharing economy has become one of the fastest growing business, with the success of Airbnb, Uber, Pace (bike sharing) and Bird (e-scooter sharing). These platforms provide new ways of accommodation and transportation. However, as users tend to act on their own interest, utility is a major problem that many businesses are facing. For example, some bike-sharing systems allow customers to drop off at any location. Though these policies best cater to the customer experience, for consistent utility in the system, companies need to commit significant resources to rebalance the bike distribution [1] or send maintenance crew for charging the e-scooters. Such large maintenance overhead drives several bike-sharing platforms to the verge of bankruptcy recently [2].

Previous research proposed to seek user cooperation with monetary incentives. Incentives are provided in mobile sensing tasks [3], [4], [5], which typically assume that users bid truthfully to execute tasks. Yet, the private cost of users is often unknown to the system. Building on the budget feasible methods [6], [7], incentives are explored in crowdsourcing tasks to learn private cost distribution and maximize utility [8]. They design fixed incentives to explore the users' costs. Incentive has been utilized to improve efficiency in the sharing economy recently. In bike sharing systems, incentives are given to the riders who are willing to cooperate and reposition their bikes to designated locations, thus rebalancing the distributions of bikes among different stations [9], [10]. Similarly, incentives determined by machine learning are offered to encourage users for taking different options such as renting an apartment with no review rating [11]. Compared with traditional algorithms, learning-based algorithms are more capable of addressing the dynamic scenarios due to human participation.

Although these works laid the foundations of incentivizing users for maximizing utility, they only consider a special case that all tasks are treated uniformly and a single distribution is learned to represent the cost profiles. In general, tasks could entail heterogeneous amount of efforts from users. For instance, in bike sharing, if there are several stations available, riders are more willing to reposition their bikes to the ones that are closer; riding to stations in further distance demands more efforts. While encouraging tenants to take different rental options, they may rank their own lists based on commute distance and safety. These *external factors* are reflected on users' choices (or implicitly, their cost for different tasks), which in turn, determine the amount of incentives to maximize the overall utility. Leveraging such context information helps learn the cost distributions more accurately. Therefore, based on the efforts required, we partition the tasks into different levels of *difficulty* and learn a cost distribution on each level. To solve this new problem, a naive solution is to invoke the mechanism of [8] independently across all levels. Yet, how to satisfy the total budget, and at the same time, maximize utility is still a difficult problem. Hence, the main challenge is to find an online budget-feasible incentivizing mechanism by considering heterogenous levels of difficulty and assigning appropriate budget for each level, such that the system utility is maximized.

To tackle this challenge, this paper studies an incentivizing system with arbitrary k levels of difficulty satisfying an overall budget. First, we derive optimal offline and online solutions with *varied* and *fixed* incentives, respectively. Then we analyze the utility ratio between these approaches in the worst case (bound of $2k$ given arbitrary budget assignment) and the case with constant bound (bound of 2 given a

- *P. Zhou and Y. Yang are with Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA. E-mail: {pengzhan.zhou, yuanyuan.yang}@stonybrook.edu*
- *X. Wei is with Department of Computer Science, Old Dominion University, VA 23529, USA. E-mail: xwei001@odu.edu*
- *C. Wang is with Department of Cybersecurity, George Mason University, VA 22030, USA. E-mail: cwang51@gmu.edu*

Manuscript received MONTH 20XX.

reasonable budget assignment). To implement the fixed incentive strategy, we propose a mechanism to determine incentives online by exploring the cost distributions from the incoming users, and dynamically allocating the budget assigned to different levels.

We notice that, in the designs of incentivizing mechanisms in the mentioned works [7], [8], [9], [11], [13], a reinforcement learning mechanism named Multi-Armed Bandits with Knapsack (MABwK) is widely adopted. This motivates us to further combine the study of generalized MAB problem with the idea of considering heterogenous levels of MABs.

In the MABwK problem [23], one of multiple arms (actions) can be chosen at the cost of some resources every time, resulting in a stochastic reward associated with each arm. The reward distributions and the mean rewards are unknown a priori. The learner expects to maximize the total rewards acquired while not exceeding the budget of any resource. The traditional MAB is a special case of MABwK consuming only resource of time, and in this paper we study the general case of MABwK[1]. Through the actions taken, the reward distributions of arms can be learned, which helps the learner to take better actions afterwards. The learner faces an exploitation-and-exploration dilemma: 1) exploiting the current knowledge and making the best decision based on them 2) exploring sub-optimal arms to acquire new knowledge and expecting more rewards in the future.

Like [7], [8], [9], [11], [13], the MAB related mechanisms can be utilized to explore the user behaviors, design reverse auction schemes, and incentivize users to participate in crowdsourcing. However, all these works only deal with one MAB system at the same time, which can not handle the case that multiple and independent MABs exist simultaneously. For example, the online multi-platform advertising needs to determine the advertising strategies across multiple independent advertisement platforms (e.g. Google, Youtube, Twitter etc.).Users' responses to the advertising of one platform are unknown beforehand, and are learned in an ongoing way via the feedbacks from the users. Different platforms with its unique group of users can be treated as different MABs. They need to be jointly considered in order to maximize the number of clicks of advertisements with given advertising budget, i.e. to minimize the cost-per-click. Similar problems are formulated as a new kind of reinforcement learning problem, which is named *k-MAB* problem in this paper.

The study of generalized k-MAB is more challenging than the design of incentivizing mechanism for the system of k-level difficulties since we can not sort each MAB according to measures like difficulties. For k-MAB problem, it is challenging to derive algorithms to find the optimal solution achieving the balance between the exploitation of current knowledge and exploration of known knowledge. In order to address this, [zz] we consider a three dimensional cost functions, calculate its optimal solution, and propose a mechanism to learn the user cost function and the best strategy.[zz]

The contributions of this paper are three-folds:

- We propose a new reinforcement learning problem named k-MAB. For its general case where the difficulties of each MAB can be sorted, we propose an online incentivizing mechanism with 2-approximation bound. The bound is also proved to be the best performance bound that any mechanism can achieve.
- We propose CUE-UCB mechanism to solve the generalized k-MAB problem. By combining the utility function and the efficiency function statistically, we propose a mechanism to find the best MAB and arm among k independent MABs for the first time. The regret analysis is given.
- We conduct a case study of electric bike-sharing and mobile advertisement click, and evaluate the proposed mechanisms on two public datasets. Compared to the previous works, the experiments demonstrate that our mechanism used for bike-sharing not only achieves about 7 times utility, but also saves 54% time to reach a utility objective. For mobile advertisement, the proposed CUE-UCB algorithm improves the utility up to 175%.

The rest of the paper is organized as follows. Section 2 studies literature. Section 3 discusses motivation and the system model. Section 4 theoretically studies the mechanisms. Section 5 proposes the k-level online incentivizing mechanism to address the special case. Section 6 studies the generalized k-MAB problem and proposes solutions. Section 7 evaluates the performance of proposed mechanism via the case study of e-bike repositioning and online advertising. Section 8 concludes the paper.

2 Related Works

2.1 Incentivizing Mechanisms

There is a plethora of literatures on the design of incentivizing mechanisms based on MAB. In [7], the authors study a basic class of mechanism design for procurement auctions, where the sellers provide varying prices to compete for the buyers. They prove the proposed budget feasible mechanisms are truthful and computational efficient, especially having an approximation ratio of two compared with the optimal. In [8], authors combine the procurement auction and multi-armed bandits to design a posted price mechanism, which achieves near-optimal utility of the requesters in the crowdsourcing tasks. They apply a regret minimization method to determine the proper incentive and prove the average regret of the mechanism approaches zero asymptotically. In [9], the authors provide monetary rewards to the users of sharing bikes who are willing to choose alternate picking or returning stations based on regret minimization. In [11], authors focus on the cold start problem in the rental platforms, which provides incentives to motivate users to deviate from their regular choices to explore some choices with rare ratings to avoid the vicious cycle of wasting viable choices. In [13], the authors construct a bipartite graph and determine the allocation of the crowdsourcing tasks by finding the match between various tasks and the workers, who have different expertise and interests. In [14], authors study the incentivizing mechanisms for a special case of MAB to achieve the maximum utility of E-bike repositioning with the user cooperation, but the work is not extended to the generalized MAB problems. In [33], a reinforcement learning mechanism is proposed based on the deep deterministic policy gradient algorithm by modeling the problem as a Markov decision process. Spatial and temporal features are used to predict the usage of sharing bikes, which maintains a divide-and-conquer structure. In [34], a deep reinforcement learning based incentive mechanism is proposed to determine the pricing strategy for the parameter server and the optimal training strategies for the edge nodes in the federated learning. It addresses the challenges of unshared information and difficulties of evaluating contributions by forming as a Stackelberg game. However, all these works do not consider the scenarios that if the users are assigned tasks with heterogenous difficulties, they expect rewards accordingly depending on the efforts they are going to make.

[1] We briefly denote MABwK as MAB hereafter for conciseness.

2.2 Multi-armed Bandits

Due to its importance of exemplifying the exploration–exploitation tradeoff dilemma, there are plenty of literature discussing various types of MAB problem. In [23], the authors discuss the general MABwK problem, which considers the one or more limited resources consumed during the learning process. However, every arm is accessible in their scenario. In [24], the case of sleeping (i.e. unavailable) bandits are jointly considered with the fairness constraints in the application like wireless scheduling. In [25], the authors intend to incentivize high quality content contributor in user generated content platform. The number of arms can increase via the process of exploring, and an incentivizing mechanism with randomization is proposed to address the issue of flooding contributions. In [26], authors take multilevel feedbacks into consideration to address the web link selection problem, which is formulated as a constrained stochastic MAB problem. In [27], authors study contextual multi-arm bandits with resource constraints to choose advertisements or design dynamic pricing, where a regret bound with square root of time horizon is derived. In [28], authors focus on scenarios where the rewards of choosing arms are binary instead of quantifiable or having natural scale, achieving information-theoretically regret bound. In [29], authors investigate stochastic and adversarial combinatorial MAB problems, and efficiently exploit the structure of the problem via the proposed algorithms under different bandit feedbacks. In [35], a Lipschitz contextual MAB problem is formulated to address the strategies of advertising based on web search of users. The mechanism derives online strategies of advertising with a guaranteed performance bound based on the given side information and the action chosen simultaneously. In [36], a non-linear deep learning framework of contextual bandits tackles the exploit-exploration trade-off by utilizing the connection between inference time dropout and the weight sampling from a Bayesian neural network. In [31], the authors design a master algorithm which adaptively selects the best algorithm among a set of base algorithm in bandit settings with superior regret bound. However, the proposed mechanism utilizes the same algorithm for each MAB in our setting. The performance gaps are not caused by the utilized algorithm itself but the intrinsic differences among independent MABs. However, in these framework, only one MAB is studied, and the available arms are all accessible to the system. We propose k-MAB problem which considers the scenarios of k independent MABs sharing the same budget and needing to be addressed concurrently.

3 Preliminary

3.1 Motivation

The previous works explore the distribution of user cost or rewards to find the optimal incentivizing strategies. Nevertheless, they assume each user or action has a private and static cost for all the tasks. In fact, one's cost is affected by many external factors, such as weather condition/walking distance (bike reposition problem),new review ratings (housing rental), or the users' interests about the product. These factors could cause the cost to vary, depending on how users perceive the task at a different time. The cost may fluctuate substantially, leading to jitters or even divergence while learning the cost distribution. If we discriminate the tasks based on the efforts needed and learn multiple cost distributions, the distributions can be approximated more accurately towards the profiles of the true cost at that states. Leveraging these context information certainly helps the system make better decisions as illustrated by the following example.

1) Sharing Bike Repositioning. A motivating example is illustrated. Consider a bike-sharing system that incentivizes users for bike rebalancing. Through marketing research and survey, the company gains some prior knowledge about the external factors with a major impact on user cost, e.g., $\{$*weather, walking distance*$\}$[2] After returning the bikes to a different station, the user may have to walk extra distance to her destination. In [9], the same incentive is provided to all users regardless of the external factors. However, during a raining day, it would be more difficult to motivate users for repositioning, thus demanding a higher incentive from the budget; when it is sunny, users are more willing to cooperate and earn rewards, thus paying a lower incentive being sufficient to avoid wasting the budget. Therefore, by considering external factors and incentivizing users accordingly, the budget can be utilized more efficiently for maximizing system utility.

There are some parallel works that assign workers to perform heterogeneous tasks [12], [13], [15]. They assume the users bid truthfully based on their cost and the system assigns tasks considering the bidding prices and the skill set of users. These problems are usually solved *offline* with known cost distribution of users, aiming to find an optimal bipartite matching between tasks and users. However, this paper studies an *online* problem that the users do not reveal their cost and the incentives are not fixed. Instead, they are learned through distributing incentives and getting response from the users.

The contextual information are not always available and the levels of difficulties may not consistently be discriminated quantitatively. However, the studied users still belong to different categories, where each category belongs to one independent MAB system. In the traditional MAB problem, the learner is faced up with one MAB system, i.e. all arms are accessible and she can choose any arm in any iteration. However, in many realistic settings, the learner is faced up with multiple independent MAB systems, and she can only make decisions within one of the MABs in each iteration. These observations motivate us to propose and study the k-MAB problem. The following two motivating examples illustrate the potential applications of this new RL problem.

2) Multi-platform Advertising. Nowadays, there are many giant IT companies in charge of multiple platforms simultaneously which can be used for advertising. Vice versa, one new product, service, or movie needs to take advantage of the multi-platform promotion to get the most exposure in the debut. One virtual advertiser is in charge of multiple platforms (e.g. blogs, video websites, news websites), each of which has a certain group of users that can be promoted with advertisements. She intends to advertise one new product with restricted advertisement buget, without knowledge of what and how users can be attracted to click on the advertisement. Within one platform, advertisements can be displayed at one of the banners (like arms in MAB) consuming different amount of budget. The success of advertising is measured via the number of clicks, while one click increases the *utility* by one. The advertiser aims to achieve the maximum utility with certain budget, i.e. the new product gets the most extensive promotion with fixed advertising budget.

There are many challenges to fulfill this objective. First, due to the stochastic patterns of incoming users, a random platform is considered each time, i.e. with limited arms to choose from. Second, the advertising actions to take should jointly consider all platforms besides of the platform being used by the incoming user. Third, the allocation of

[2] Due to space limit, this paper does not attempt to come up with an exhaustive list of external factors for specific applications. However, the proposed mechanism would work with more factors once they are determined from data analytics.

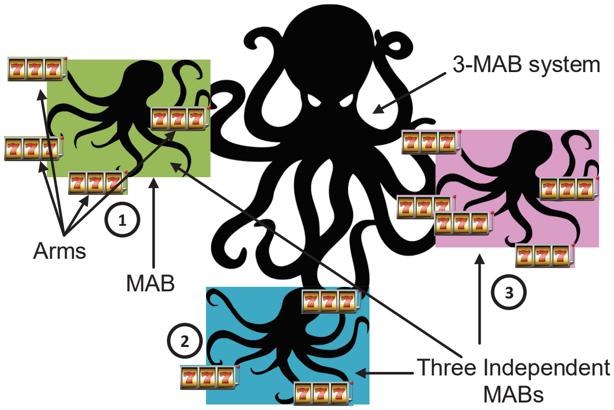


Fig. 1: An example of 3-MAB system.

budgets to any platform affects the utility achieved via other platforms, which in turn affects the total utility achieved. A natural idea is to push more advertisements to users more likely to be attracted by the product while reduce the advertisements displayed to the ones lacking of interests.

3) Multi-casino Gambling. The MAB problem is usually exemplified in the example of gambling in a casino, where the arms of various slot machine can be pushed to gain potential rewards at the cost of bets. Similarly, one gambler plays online gambling in multiple casinos simultaneously. The game in each casino is modeled uniquely by a MAB, i.e. the gambler stochastically gets some rewards while putting some bets at a slot machine in each game. She has no knowledge about the distribution of rewards at the beginning. Since each game takes some time, the next game starts randomly in one of the casinos for the gambler. Note that, the acquired rewards can be put back for another gambling. She aims to get the most money with certain initial funds within the given time period. In this example, the gambler needs to distribute her funds playing in each casino based on the learned distributions of each game, while the game in each casion is an independent MAB. Due to the nature of gambling, the bet of each action can be adjusted, and she can get more rewards with larger bets if she wins. Thus, the bets she puts in one game may affect the bets she can put in other games in the future. The optimal strategy to play the multi-casino gambling to get the most rewards is extremely challenging.

Note that, the gambling is just utilized here to better illustrate the possible application of the studied novel problem, since gambling-like scenarios often take place in realistic settings. E.g. the clinical trails of vaccine or medication need to investigate the effects of different treatments while minimizing the patient or time lost [16]. The trail of one kind of medication can be treated as one MAB while a pharmaceutical company may need to test several potential medication simultaneously to find the best, which is constrained by limited resources of the company. Especially in face of a global pandemic like COVID-19, the investment of the world-wide development of vaccine is like the multi-casino gambling.

3.2 System Model

We first discuss about the user model. The users are treated as a group instead of individuals in the system, whose behavior as a whole is intended to be learned rather than individual's behaviors to protect privacy. At any time t, random number of users arrive in the system, and the system assigns an offer of task to the arriving user via the trained mechanism. The posted price mechanism is adopted here to reduce the time spent for task assignment, enhancing the user experience. The user has only one chance to either accept or decline the offer, and she only receives incentive if she accepts the offer. Next, we formally define the proposed k-MAB problem.

Definition 1. *k-Multi-Armed Bandits (k-MAB). Each time, the learner is randomly given one MAB from a set of k independent MABs to play with, while consuming some budget corresponding to the chosen arm. The objective is to get the maximum rewards when the budget is depleted.*

The proposed k-MAB problem is demonstrated via an example of 3-MAB in Fig. 1. A classic illustration of MAB problem is an octopus playing slot machines via its tentacles, where each slot machine corresponds to a stochastic reward of certain distribution waiting for exploring. As shown in Fig. 1, three little octopuses denoted by different colors represent three independent MABs. The giant octopus can control the three little octopuses, hence playing different MABs. The giant octopus can only play one of the MABs at once consuming some budget, intending to maximize its rewards when depleting the budget.

Note that, the general *MAB* problem is a special case of the *k-MAB* problem for $k = 1$ and all the arms are accessible to the system at any time. Since the budget spent for one of the k MABs will inevitably affect the left budget spent for the other $k - 1$ MABs, henceforth affecting the total utility achieved with the given budget, the solving of k-MAB is extremely challenging. Therefore, we start with the special case that these k MABs are sorted in an order of difficulties. C is the spent budget for a user, and different C's associated with different incentivizing levels are the arms in the MAB problem. The expected reward for an MAB denoted by i with spent budget C is equal to $r_i(C)$. If one MAB i is more difficult than the other MAB j, then with the same amount of spent budget, the reward of i is always no larger than the reward of j. The k MABs are sorted in the ascending order of the difficulty (i.e. i+1-th MAB is always more difficult than the i-th MAB), which is shown in the following,

$$r_i(C) \geq r_j(C), \forall i \leq j, \forall C. \tag{1}$$

We first discuss this special case and provide a performance guaranteed solution via demonstrating its applications in the sharing economy.

Definition 2. *Task difficulty. Each level of difficulty is defined by a point in the space of external factors.*

In order to address the non-stationarity of users, the external factors are utilized to depict different situations. E.g. the tasks under different weather conditions are classified into different levels to be learned specifically. Including more factors, the classified levels are more stationary in the trade-off of fewer users in each level to be learned. Therefore, the two main factors of weather and walking distance are picked in the simulation. We may conduct similar processes for multi-platform advertising to address non-stationarity via market survey in advance.

With n external factors, the i-th factor has m_i levels. The total k levels of difficulty are represented as a product from all the levels, $k = \prod_{i=1}^{n} m_i$. E.g., $\{\{raining, sunny\}, \{< 500m, \geq 500m\}\}$ for the factors of weather and walking distance in bike-sharing systems ($k = 4$).

The system has certain budget to incentivize the users to accomplish an objective, which consists of tasks with varied levels of difficulty. When a user arrives, the system determines the difficulty of completing the task according to the current situation. For instance, on a raining day, a station within 500m needs reposition. An incentive is determined based on the cost distribution learned online at that level. The user either accepts the offer if the incentive is no less than her cost, or declines if it is deficient. Our strategy is a posted price mechanism that ensures truthfulness by making the offered incentive independent of the cost claimed by the user [17], [18]. Instead of building on truthful bidding/auction mechanisms such as second-price auction [19], the posted price mechanism is adopted here due to: 1) users may not intend to reveal their intrinsic cost due to privacy; 2) system handles incoming requests one by one and an immediate decision is made; 3) if we were to use auction, the system should maintain a time interval to gather enough users, and establish interactive sessions for the bidding process, which hurts the user experience.

Definition 3. *k-level (incentivizing) system. Tasks have k levels of difficulty. A user can conduct only one task at a certain level. The cost in the system for the j-th user to finish the task at the i-th level, $C_j^{(i)}$ is sorted in an ascending order, $C_1^{(i)} \leq C_2^{(i)} \leq \ldots \leq C_{n_i}^{(i)}$. n_i is the number of users that perform the tasks at the i-th level. The difficulties of the tasks are also arranged in an ascending order, i.e. the $(i+1)$-th level is more difficult than the i-th level.*

According to *Definition 3*, we naturally assume that the cost in level $i+1$ is larger than the cost in level i for the same position j in the sorted list,

$$C_j^{(i)} < C_j^{(i+1)}, \ \forall\, j \text{ and } \forall\, 1 \leq i \leq k-1. \tag{2}$$

Definition 4. *Utility. The summation of the number of tasks completed by the users in each level via the incentivizing mechanism.*

According to the above definition, utility is equal to the number of users willing to accept the offers if users accepting the offers are obligated to complete the tasks. In order to assure this, a reputation system may be built to measure the reliability of the users. Users with low reputation are barely considered to receive any more offers in the future. The construction of the reputation system is beyond the scope of this work. Note that, the extreme case where users are unable to fulfill the assigned tasks due to personal issues is not considered in the paper.

Definition 5. *Budget feasibility. With a total budget B, B_i is the portion to be assigned to the i-th level. Their sum should be within the total budget, $\sum_{i=1}^{k} B_i \leq B$, and for any i, the total incentives provided by any mechanism to the i-th level should not exceed B_i.*

The system has sufficient participants, $n_i \geq B_i/C_1^{(i)}$ for each level, to make sure that all the budgets are utilized. The number of participants is finite; otherwise, it would be a trivial problem since we can simply assign the minimum incentive to each user but still find enough participants.

4 Mechanism and Analysis

The goal is to design a truthful, budget-feasible mechanism that achieves a constant approximation ratio to the optimal solution. There are two strategies of assigning incentives.

Definition 6. *OPT-VAR. The optimal solution which achieves the maximum utility for the k-level system by providing varied incentives to each user.*

Definition 7. *OPT-FIX. The optimal solution which achieves the maximum utility for the k-level system by providing fixed incentives to each user at the same level.*

We discuss how OPT-VAR and OPT-FIX are achieved in the following lemmas.

Lemma 1. *OPT-VAR is achieved by sorting the cost of all users in an ascending order and providing incentives in the sorted order until the budget is exhausted.*

Proof. Prove by contradiction. Assume a budget-feasible solution that achieves larger utility, but the cost does not follow the sorted order, i.e., there must exist one user with lower cost who is not chosen, but the one with higher cost has been chosen. Then there is always a solution that maintains the utility and budget feasibility by switching the user of higher cost with the one of lower cost (that are not chosen), which still follows the sorted order of the cost. It is an obvious contradiction to the assumption, so the lemma is proved. □

Lemma 2. *OPT-FIX can be achieved by providing the fixed incentive $C_{q_i}^{(i)}$ to the first q_i users in the i-th level, where q_i is the largest number such that $C_{q_i}^{(i)} \leq \frac{B_i}{q_i}$, $\forall 1 \leq i \leq k$.*

Proof. For any $1 \leq j \leq q_i$, since $C_j^{(i)} \leq C_{q_i}^{(i)}$, providing $C_{q_i}^{(i)}$ ensures that the first q_i users would accept the task. Meanwhile, $C_{q_i}^{(i)} \cdot q_i \leq B_i$ makes the mechanism budget-feasible.

Optimality can be proved by contradiction as well. Assume OPT-FIX is larger than the utility achieved by this mechanism, there must be at least one $q_i' > q_i$ such that $C_{q_i'}^{(i)} \cdot q_i' \leq B_i$. However, q_i is the largest number satisfying $C_{q_i}^{(i)} \leq \frac{B_i}{q_i}$ for budget feasibility, thereby causing a contradiction. The lemma is proved. □

The derivation of B_i is one of the main contributions of this work, which is discussed in Theorem 2. For the same level of difficulty, OPT-FIX provides fixed amount of incentives. It is certainly not as efficient as OPT-VAR since the incentives provided may exceed the actual cost of users. Therefore, the utility of OPT-FIX cannot surpass OPT-VAR. However, OPT-VAR requires all the cost to be known, so more suitable for planning offline. Most platforms take streaming requests and make decisions online. To this end, we pursue *fixed incentive* as an online approach and find the ratio between OPT-FIX to OPT-VAR for the k-level system[3]. Budget assignment among all k levels is a difficult problem since the cost distribution on each level is unknown. To start, consider an arbitrary budget assignment below.

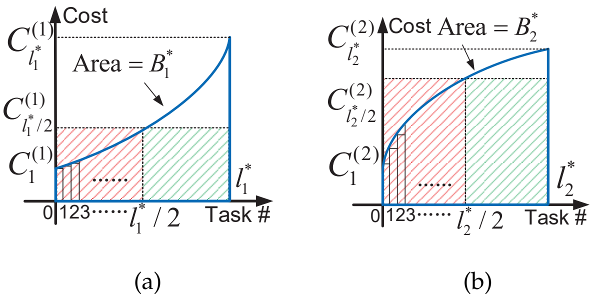
Theorem 1. *For the k-level system, OPT-VAR $\leq 2k \cdot$ OPT-FIX, i.e. $l^*(k) \leq 2k \cdot l(k)$, for any distribution of user cost with arbitrary budget assignment of B_i for any $1 \leq i \leq k$. $l^*(k)$ and $l(k)$ are the utility of OPT-VAR and OPT-FIX for the k-level system respectively.*

Proof. We prove this theorem by mathematical induction.
Base case: The work of [7] has proved this base case when $k=1$ (only one level of difficulty).
Inductive step: For $k \geq 1$, assume that $l^*(k) \leq 2k \cdot l(k)$ holds, we want to prove that $l^*(k+1) \leq 2(k+1) \cdot l(k+1)$ also holds, where in addition to the k levels, one new level is added with a total of $(k+1)$ levels in the system.

The difference between $l^*(k+1)$ and $l^*(k)$ is denoted as $\Delta l^* = l^*(k+1) - l^*(k)$. Rewrite this into, $l^*(k+1) = l^*(k) + \Delta l^*$. To prove $l^*(k+1) \leq 2(k+1) \cdot l(k+1)$, it is sufficient to prove that both 1): $l^*(k) \leq 2k \cdot l(k+1)$ and 2) $\Delta l^* \leq 2 \cdot l(k+1)$ hold.

1) We prove $l^*(k) \leq 2k \cdot l(k+1)$. Introducing the new $(k+1)$-th level means more options that users can choose from (i.e., higher chances for the incentives to get accepted). Compared with the k level system, the new $k+1$-th level is added to the $k+1$ level system. If the assigned budget to the $k+1$-th level is set to 0, the found optimal solution is the same as the optimal solution found for the k level system. Hence, the utility of $(k+1)$-level system is at least as good as the k level: $l(k) \leq l(k+1)$. Plug in this into baseline assumption $l^*(k) \leq 2k \cdot l(k)$, then $l^*(k) \leq 2k \cdot l(k+1)$ is proved.

[3] For simplicity, OPT-VAR and OPT-FIX also stand for utility achieved by the mechanisms henceforth.



(a) (b)

Fig. 2: Cost of users in the incentivizing system when $k = 2$ (a) level 1 tasks (b) level 2 tasks. Task # = 1 means the 1st task of level 1, with its incentive of $C_1^{(1)}$ provided by OPT-VAR. The area of the rectangular bar for each task is the budget required to complete that task and their sum equals to the total budget B_1^* or B_2^*.

2) We prove $\Delta l^* \leq 2 \cdot l(k+1)$. The sketch is to apply Eq. (2), which implies that the cost in the $(k+1)$-th level is larger than the k-th level for the same position j in the sorted list. Since the cost is relatively higher at $(k+1)$-th level, the number of tasks that can be successfully performed is no greater than that from level k, i.e. $l^*_{k+1}(k+1) \leq l^*_k(k)$. Similarly, for k levels, $l^*_k(k) \leq l^*_i(k)$ and $l^*(k) = \sum_{i=1}^{k} l^*_i(k)$, from which it can be derived that $l^*_k(k) \leq \frac{l^*(k)}{k}$. That is, the tasks that can be achieved at the k-th level are no greater than the average number of tasks achieved at each level, because the k-th level is the most difficult. Then from the upper bound of Δl^*, the second condition is proved as,

$$\Delta l^* \leq l^*_{k+1}(k+1) \leq l^*_k(k) \leq \frac{l^*(k)}{k} \leq 2 \cdot l(k) \leq 2 \cdot l(k+1). \quad (3)$$

Both 1) and 2) are proved so the ratio of $2k$ is proved. □

Theorem 1 states that an arbitrary budget assignment among all the levels can still achieve a bounded ratio of $2k$ proportional to fixed k. The next question is to what extent OPT-FIX can achieve compared to OPT-VAR (constant approximation ratio). To find such ratio, we assume an optimal budget assignment is given such that running this assignment the budget will be fully utilized at each level without causing an overall budget-infeasibility. We start with the special case of $k = 2$ and extend it into the general case. The cost of level 1 and level 2 tasks are sorted as, $C_1^{(1)} \leq C_2^{(1)} \leq \ldots \leq C_{n_1}^{(1)}$ and $C_1^{(2)} \leq C_2^{(2)} \leq \ldots \leq C_{n_2}^{(2)}$. The total budget is B, in which B_1 is reserved for the level 1 tasks and B_2 for the level 2 tasks ($B = B_1 + B_2$). The budget assigned by OPT-VAR are B_1^* and B_2^* ($B = B_1^* + B_2^*$). The total number of the tasks assigned by OPT-VAR is denoted by l^*, where l_1^* and l_2^* are the numbers of level 1 and level 2 tasks respectively ($l^* = l_1^* + l_2^*$)[4]. Similarly, OPT-FIX assigns l tasks in total, where $l = l_1 + l_2$. In the following, we prove that the ratio of OPT-VAR and OPT-FIX is bounded by 2.

Lemma 3. *When $k = 2$, OPT-VAR $\leq 2 \cdot$ OPT-FIX for any distribution of user cost if $B_1 \geq \frac{l_1^*}{2} \cdot C^{(1)}_{l_1^*/2}$, $B_2 \geq \frac{l_2^*}{2} \cdot C^{(2)}_{l_2^*/2}$, and $B_1 + B_2 \leq B$. There are always such B_1 and B_2 satisfying these constraints simultaneously. $\frac{l_i^*}{2}$ $(i = 1, 2)$ denotes half of the tasks determined by OPT-VAR.*

Proof. The proof is illustrated with the help of Fig. 2, in which (a) and (b) show sorted lists of tasks vs. their ascending cost using OPT-VAR. We focus on level 1 and the same principle follows for level 2. Connecting the costs from $C_1^{(1)}$ to $C^{(1)}_{l_1^*}$ results the cost curve in blue. Because of adequate user participation, the sum of all the rectangular bars can be closely approximated by the integral of the cost curve (the area beneath it).

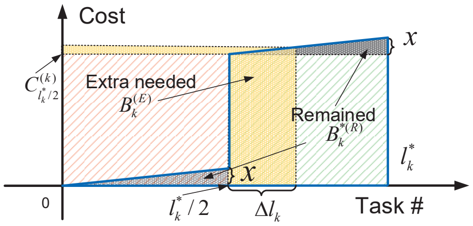


Fig. 3: Counterexample: cost distribution violating the $2 - \epsilon$ approximation bound.

If $B_1 = \frac{l_1^*}{2} \cdot C^{(1)}_{l_1^*/2}$, OPT-FIX can provide incentive $C^{(1)}_{l_1^*/2}$ to all first $l_1^*/2$ users in level 1. Its budget is represented by the area in red (Fig. 2 (a)). Because of $\frac{l_1^*}{2}$ is the mid-point, the area of the red and green rectangles are the same. Thus, by flipping the red into the green area, it is easy to see that $B_1 = \frac{l_1^*}{2} \cdot C^{(1)}_{l_1^*/2} \leq B_1^*$, and the utilities l_1 achieved by B_1 via the OPT-FIX mechanism satisfies $l_1 = \frac{l_1^*}{2}$. Similarly, for level 2, if $B_2 = \frac{l_2^*}{2} \cdot C^{(2)}_{l_2^*/2}$, the utilities l_2 achieved by B_2 via OPT-FIX satisfies $l_2 = \frac{l_2^*}{2}$ and $B_2 \leq B_2^*$. Thus, $l_i = \frac{l_i^*}{2}$, for $i = 1, 2$. The relation holds for both concave and convex curves since it only relies on the first derivative of the curve (monotonically increasing), but not the second derivative. When B_1 and B_2 are chosen as above, $B_1 + B_2 \leq B_1^* + B_2^* = B$, and this budget-feasible mechanism infers, $l_1 + l_2 = \frac{l_1^*}{2} + \frac{l_2^*}{2} \Longrightarrow l = \frac{l^*}{2}$ □

Lemma 3 can be conveniently extended for the general case of k as discussed in the next theorem.

Theorem 2. *For the incentivizing system of k levels, OPT-VAR $\leq 2 \cdot$ OPT-FIX for any distribution of user cost if $B_i \geq \frac{l_i^*}{2} \cdot C^{(i)}_{l_i^*/2}$, and $\sum_{i=1}^{k} B_i \leq B$. There are always such B_i satisfying these constraints simultaneously.*

Proof. According to *Lemma 3*,

$$B_i \geq \frac{l_i^*}{2} \cdot C^{(i)}_{l_i^*/2} \Longrightarrow l_i \geq \frac{l_i^*}{2}. \quad (4)$$

For budget feasibility, B_i should satisfy $\sum_{i=1}^{k} B_i \leq B$ and,

$$l_i \geq \frac{l_i^*}{2} \Longrightarrow \sum_{i=1}^{k} l_i \geq \sum_{i=1}^{k} \frac{l_i^*}{2} \Longrightarrow l \geq \frac{l^*}{2}. \quad (5)$$

Such B_i always exists by simply setting $B_i = \frac{l_i^*}{2} \cdot C^{(i)}_{l_i^*/2}$ for any i. The relations in Fig. 2 still hold for any level i of the k-level system,

$$\frac{l_i^*}{2} \cdot C^{(i)}_{l_i^*/2} \leq B_i^* \Longrightarrow B_i \leq B_i^* \Longrightarrow \sum_{i=1}^{k} B_i \leq B. \quad (6)$$

□

Theorem 3. *For any k-level system and $\epsilon > 0$, there is always a distribution of user cost such that OPT-VAR $> (2-\epsilon) \cdot$OPT-FIX.*

Proof. For $\epsilon > 1$, the theorem obviously holds since OPT-VAR $\geq$ OPT-FIX due to the inefficient use of budget of OPT-FIX. For $0 < \epsilon \leq 1$, we prove by contradiction. Assume there exists such ϵ that OPT-VAR $\leq (2 - \epsilon) \cdot$ OPT-FIX, i.e.

[4] We omit the (k) notation here for clarity since we refer to the k-level system hereafter.

$l \geq \frac{l^*}{2-\epsilon}$ according to (5). *Theorem 2* has shown that l of OPT-FIX should reach $\frac{l^*}{2}$ to satisfy budget feasibility ($l = \frac{l^*}{2}$). The increment of Δl over l should satisfy, thus the increment Δl of OPT-FIX beyond $\frac{l^*}{2}$ should at least satisfy,

$$l + \Delta l \geq \frac{l^*}{2-\epsilon} \Longrightarrow \Delta l \geq \frac{l^*}{2-\epsilon} - \frac{l^*}{2} = \frac{\epsilon}{2(2-\epsilon)} \cdot l^*. \qquad (7)$$

For any given $0 < \epsilon \leq 1$, $\frac{\epsilon}{2(2-\epsilon)}$ is a positive number ($<$ 0.5). Since this relation should hold for any distribution of costs, we can construct the following distribution of a k-level system,

$$\sum_{i=1}^{k-1} n_i < \frac{\epsilon}{2(2-\epsilon)} \cdot l^* - 1, \qquad (8)$$

$$C^{(i)}_{n_i} \leq x, \; \forall\, 1 \leq i \leq k-1, \qquad (9)$$

where n_i is the total number of users participating in the level i tasks, x is an adjustable parameter to determine the distribution of costs. The cost distribution of level k is depicted by the blue curve in Fig. 3. The OPT-VAR for this distribution of costs can maintain l^*, by adjusting l_k^*.

Δl_i is the increment of l_i of OPT-FIX. Since $\Delta l = \sum_{i=1}^{k-1} \Delta l_i + \Delta l_k$, and (7),

$$\sum_{i=1}^{k-1} \Delta l_i \leq \sum_{i=1}^{k-1} n_i < \frac{\epsilon}{2(2-\epsilon)} \cdot l^* - 1 \Longrightarrow \Delta l_k > 1. \qquad (10)$$

As shown in Fig. 3, we proved in *Theorem 2* that the budget corresponding to the red shadow area can be covered by the green shadow area. In order to cover the budget required by $\frac{l_k^*}{2} + \Delta l_k$, OPT-FIX needs the extra budget $B_k^{(E)}$ denoted by the yellow shadow area, and the remained budget of B_k^* besides of the green shadow area is $B_k^{*(R)}$ denoted by the black shadow area in the figure. The total remained budget of the whole system is $B^{(R)}$,

$$B^{(R)} \leq \sum_{i=1}^{k-1} B_i^* + B_k^{*(R)}. \qquad (11)$$

Since $B_i^* \leq l_i^* \cdot C^{(i)}_{n_i} \leq l_i^* \cdot x$, $B_k^{*(R)} \leq \frac{l_k^*}{2} \cdot x$, and $l^* = \sum_{i=1}^{k} l_i^*$, the upper bound of the remained budget is,

$$B^{(R)} \leq (\sum_{i=1}^{k-1} l_i^*) \cdot x + \frac{l_k^*}{2} \cdot x < l^* \cdot x. \qquad (12)$$

Meanwhile, the extra budget needed should satisfy,

$$B_k^{(E)} \geq C^{(k)}_{l_k^*/2} \cdot \Delta l_k > C^{(k)}_{l_k^*/2}. \qquad (13)$$

Let x be any number s.t. $0 < x < C^{(k)}_{l_k^*/2}/l^*$, thus,

$$B^{(R)} < l^* \cdot x < C^{(k)}_{l_k^*/2} < B_k^{(E)}. \qquad (14)$$

$B^{(R)} < B_k^{(E)}$ means that the remained budget of the whole k-level incentivizing system is not larger than the extra budget needed in order to increase l_k by Δl_k. $B^{(R)} < B_k^{(E)}$ means that the OPT-FIX is not budget-feasible, thus such OPT-FIX does not exist for this constructed distribution of costs. This leads to the contradiction of the assumption that there exists an $0 < \epsilon \leq 1$ such that OPT-VAR $\leq$ $(2-\epsilon) \cdot$ OPT-FIX for any distribution of costs. Therefore, it is proved that $\forall \epsilon > 0$, there is always a distributions of users' costs such that OPT-VAR $>$ $(2-\epsilon) \cdot$ OPT-FIX. The ratio 2 in *Theorem 2* is a tight bound. □

5 *k*-Level Incentivizing Mechanism

We implement OPT-FIX under the framework of multi-armed bandit (MAB). In MAB, the learner pulls an arm each time and receives a stochastic reward. To maximize the reward, she needs to exploit the best arm, and meanwhile, explore other potentially optimal arms. We map the MAB framework to the k-level incentivizing system. Here, the goal is to learn users' cost distributions by minimizing the regret, which is the difference between the expected and actual utility from a chosen incentive [20], [21]. By dynamically adjusting the budget assignment according to learned cost profiles, we want to minimize the regrets across all the levels. The mechanism is described below.

Users randomly arrive at the system one at a time. Based on the external factors under the current setting, the incoming user is dispatched to a desired difficulty level i. For example, repositioning the bike to a station with shortage at 500m distance in a sunny day. The system distributes incentive v_i to the user according to the current cost distribution at the i-th level (discussed next). The user compares the incentive with her private cost and responds either “accept” or “decline” to the system. The system then updates the cost distribution for this level based on the response; the proportion of budgets assigned to each level is adjusted according to the new cost distribution. These steps are repeated until the budget is depleted. Specifically, the incentive $v^{(i)}$ for the incoming request in level i is,

$$v^{(i)} = \operatorname*{arg\,max}_{C^{(i)}_{\text{min}} \leq v \leq C^{(i)}_{\text{max}}} \min \left\{ \frac{B_i}{v}, P_i(v) \cdot n_i \right\}. \qquad (15)$$

v is a discrete variable in the range of $C^{(i)}_{\text{min}}$ and $C^{(i)}_{\text{max}}$, which are the minimum and maximum incentives allowed in level i. $\frac{B_i}{v}$ is the number of tasks that can be completed with budget B_i by running with incentive v. n_i is the number of users performing level i tasks. $P_i(v)$ is the probability that the randomly arriving user accepts the level i task for the incentive v according to the learned distribution for level i. $P_i(v) \cdot n_i$ is the expected number of users who would accept the tasks given incentive v at level i. The minimum of the two numbers is the actual number of tasks that can be completed given v and the system searches for incentive $v^{(i)}$ that maximizes the number of tasks being accepted.

The number of users willing to accept the offers in each levels are estimated via the Line 14 in Algorithm 1. Sort these levels in an ascending order, and incentivizing users accordingly derives the required budget for each level. Repeat this process until the OPT-VAR is found when B is exhausted. Based on OPT-VAR, the l_i^* can be derived, further determining B_i's.

Finding the optimal budget assignment for the maximum utility turns out to be difficult (at least in the NP category). For computational efficiency, we pursue the direction of approximation derivations and use them as a guideline for the learning mechanism. *Theorem 2* states that by assigning budget $B_i = \frac{l_i^*}{2} \cdot C^{(i)}_{l_i^*/2}$ to level i, the 2-approximation ratio is achieved. l_i^* is found by sorting incentives in an ascending order, and providing incentives in that order until the overall budget B is exhausted. Since the learned distributions of users' cost vary over time, B_i is updated accordingly at each level i. To fully utilize B, B_i is scaled by the factor of $\beta = B / \sum_{i=1}^{k} B_i$. The mechanism is summarized in Algorithm 1 and evaluated next.

6 Mechanisms for k-MAB

In Section 4 and 5, we solve the k-MAB problem with the known k levels of difficulty requiring the satisfied condition of Eq. (1). The k-level online incentivizing mechanism with

Algorithm 1: *k*-level online incentivizing mechanism

```
Input: # of levels $k$, total budge $B$, number of users $n_i$ for level $i$, min and max allowed incentive $C^{(i)}_{\text{min}}$ and $C^{(i)}_{\text{max}}$ for level $i$, incentive increment $\Delta v$, set of incoming users $\mathcal{U}$.
Output: Incentive $v^{(i)}$ for incoming users at level $i$.
$S \leftarrow 0, S_i \leftarrow 0, N^{(i)}_j \leftarrow 0, l \leftarrow 0, l_i \leftarrow 0, \forall i, j$
for $\forall\ u \in \mathcal{U}$ do
    Determine the level $i$ that $u$ belongs to
    $v^{(i)}_j \leftarrow C^{(i)}_{\text{min}} + (j-1) \cdot \Delta v, \forall j$
    $v^{(i)} \leftarrow \underset{C^{(i)}_{\text{min}} \leq v^{(i)}_j \leq C^{(i)}_{\text{max}}}{\arg\max} \min\{\frac{B_i}{v^{(i)}_j}, P_i(v^{(i)}_j) \cdot n_i\}$
    if $S_i + v^{(i)} \leq B_i$ then
        Provide $v^{(i)}$ to $u$, and collect her response $r_u$
        $P_i(v^{(i)}) = P_i(v^{(i)}) + \frac{r_u - P_i(v^{(i)})}{N^{(i)}_j + 1}$,
          $N^{(i)}_j \leftarrow N^{(i)}_j + 1$
        $l \leftarrow l + r_u, l_i \leftarrow l_i + r_u$ //Update utility
        if $r_u = 1$ then
            $S_i \leftarrow S_i + v^{(i)}$ //The total used budget
    $n^{(i)}_j \leftarrow n_i(P_i(v^{(i)}_j) - P_i(v^{(i)}_{j-1})),\ \forall\ i, j$
    Sort $v^{(i)}_j$ in an ascending order, getting sequence $\mathcal{V}$
    while $S < B,\ \&\ \mathcal{V} \neq \phi$ do
        Extract level $i$ and order $j$ from $V_1$ //Finding $l^*_i$
        if $S + n^{(i)}_j v^{(i)}_j < B$ then
            $S \leftarrow S + n^{(i)}_j v^{(i)}_j,\ l^*_i \leftarrow l^*_i + n^{(i)}_j,\ \mathcal{V} \leftarrow \mathcal{V} \setminus V_1$
    $B_i \leftarrow \frac{l^*_i}{2} \cdot C^{(i)}_{l^*_i/2}, \forall i$ //According to Theorem 1
    $\beta \leftarrow B / \sum_{i=1}^{k} B_i$; $B_i \leftarrow B_i \cdot \beta,\ \forall i = 1, 2, \ldots, k$
    $\mathcal{U} \leftarrow \mathcal{U} \setminus u$ //Remove $u$, and process the next user
```

guaranteed performance bound 2 compared with the optimal is proposed and proved. In this section, the generalized k-MAB problem without the prerequisite of the known difficulty is studied, and a mechanism named CMUE-UCB is proposed to address this general case. The contextual bandits algorithms may not address our problem. No contextual information is available when the k-MAB problem is addressed. Instead, k independent MABs are learned separately via the user feedbacks from different MABs.

The removal of the condition of known difficulty level of the k MAB system (i.e. Eq. (1)) brings more challenges. The proofs of Theorem 1, 2, and 3 all depend on the existing of Eq. (1), i.e. the k known MABs can be strictly sorted according to their difficulty. By knowing Eq. (1), it is assured that the rewards of providing the same incentive (arm) to users of more difficult MABs can never be larger than the rewards of easier MABs, which ensures the existing of theoretical performance bounds. For the generalized k-MAB problem without known levels of difficulty, utilizing the previous mechanism may lead to the failure of finding the optimal arm or getting stuck in some sub-optimal MABs, which wastes the budget significantly. These observations motivate us to propose new mechanisms to solve the generalized k-MAB problem.

To illustrate the problem and the mechanism clearly, the k-MAB is studied with the example of the multi-platform advertising mentioned in Section 3. An advertiser needs to realize the utmost promotion of a product via serving advertisements in multiple platforms (MABs) with a given budget. Each platform has different banners (arms) charging different amount of money, which have different distributions of the possibility that the user will click on the advertisement via the banner. The advertiser only pays certain money if the user of the platform clicks on the advertisement [30].

The choice of banner in each platform is a MAB problem, and the k independent platforms form a k-MAB problem. The i-th platform is denoted by i, which has a known number of n_i users. The optimal advertising banner (arm) v determined for an incoming user should jointly consider the budget B and the click-through possibility $P(v,i)$. [5] The choice of the more expensive banner results in the fewer number of the served advertisement with given budget while likely (not necessarily) larger possibility that the user will click on the advertisement. Note that the limited number of available users n_i of the platform (MAB) i prevents the advertiser from always choosing the cheapest banner, otherwise, the advertising will fail without attracting enough users to click on within the given period. B/v represents the allowed number of advertisements served by exhausting the budget; $P(v,i) \cdot n_i$ represents the number of users clicking on the advertisment. For given v and i, the smaller one of B/v and $P(v,i) \cdot n_i)$ determines the actually realized utility (i.e. the number of the clicks of the advertisement). The objective is to find the optimal banner v and the platform i based on the current knowledge of $P(v,i)$, which realizes the maximal utility. The problem is formulated as following.

$$\mathbf{P1}: \max_{i}\{\arg\max_{v} \min\{\frac{B}{v}, P(v,i) \cdot n_i\}\}. \tag{16}$$

v is determined first by solving $\arg\max_{v} \min\{\frac{B}{v}, P(v,i) \cdot n_i\}$, which derives a utility function of i. Finding the platform i corresponding to the maximal utility tells the kind of users that should be advertised. Solving Eq. (16) gives the optimal solution based on the current knowledge of the cost distribution. However, always choosing the banner according to the solution of P1 may not help the system achieve the best performance for two reasons:

1) Exploitation-and-Exploration: At the beginning, the probability $P_i(v)$ of the user in platform i clicking on the advertisement is unknown to the advertiser, [6] which needs to be updated based on the user response r_u using the following formula,

$$P^{t+1}_i(v) = P^t_i(v) + \frac{r_u - P^t_i(v)}{N^{(i)}_v + 1}. \tag{17}$$

t denotes the time, which increases by one with one incoming user. $N^{(i)}_v$ represents the number of users that have been served with advertisements. For each platform, the banner $v^{(i)}_l$ and the maximum utility $l_t(i)$ that platform i can achieve are determined in the following equations,

$$v^{(i)}_l = \underset{C^{(i)}_{\text{min}} \leq v \leq C^{(i)}_{\text{max}}}{\arg\max} \min\{\frac{B}{v}, P_i(v) \cdot n_i\}, \tag{18}$$

$$l_t(i) = \min\{\frac{B}{v^{(i)}_l}, P_i(v^{(i)}_l) \cdot n_i\}. \tag{19}$$

[5] We use $P(v,i)$ instead of $P_i(v)$ here to emphasize that i is also a variable affecting the probability.

[6] $P(v,i)$ is denoted as $P_i(v)$ hereafter for conciseness.

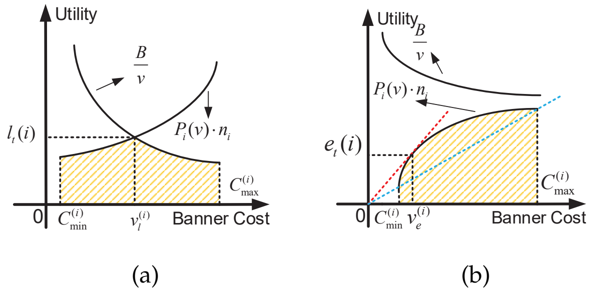


Fig. 4: Determination of the optimal banner based on utility and efficiency (a) Banner determined by utility function (b) Banner determined by efficiency in case of inadequate users.

C_{min} and C_{max} represent the cheapest and the most expensive banners in platform i. With the growing knowledge of $P_i(v)$, the optimal solution found based on current knowledge may not be the global optimal solution. In order to explore the potential more rewarding banners, some sub-optimal banners in the current stage should also be considered. The Upper Confidence Bound (UCB) can quantify the horizon of the exploration and sub-optimality gaps, which overcomes the limitation of the strategies only based on exploitation of current knowledge. The UCB depicts the extent of the uncertainty, which shows the extent of the deviation from the mean payoffs $l_t(i)$ of one arm based on the plausible possibility, which can be derived by calculating the UCB regret [8]. Combining the mean payoffs together with the UCB, the proposed algorithm determines the next platform that is served with the advertisement via the following formula,

$$\arg\max_i (l_t(i) + a\sqrt{\frac{\ln t}{M_t(i)}}), \tag{20}$$

where a is a positive user input parameter determining the contribution of the mean payoffs and the UCB, i.e. the emphasis on exploitation or exploration. $M_t(i)$ denotes the times of the advertisements having been served to the users in platform i at time t. If a is larger, then the advertiser is willing to try more sub-optimal banners to find the potential optimal; while smaller a means the advertiser tends to exploit the current knowledge more.

2) Inadequate Users: Only finding the appropriate banner according to Eq. (20) may fail in the realistic settings, since sometimes there is inadequate number of users in platform i. If the curves of B/v and $P_i(v) \cdot n_i$ have an intersection within the range of $(C_{\text{min}}^{(i)}, C_{\text{max}}^{(i)})$ as shown in Fig. 4 (a), then Eq. (18) can find the optimal banner for platform i. The yellow shadow area denotes the utility that can be achieved, and $l_t(i)$ is the peak of the area which corresponds to the maximal utility that can be achieved for platform i. If the two curves do not have any intersection as shown in Fig. 4 (b), then there are inadequate number of users to be advertised, and the condition $P_i(C_{\text{max}}^{(i)}) \cdot n_i \leq B/C_{\text{max}}^{(i)}$ is satisfied. In this case the platforms with inadequate users can not compete with other platforms with abundant users since the values of their utility functions are lower. This may result in inefficient exploration of those minor platforms, although the users of those platforms may be more enthusiastic about the advertisement and have larger probability $P_i(v)$ to click on the advertisement with the same v compared with other major platforms. In order to maximize the utility, another mechanism based on the efficiency of the advertisement is proposed to provide a way to explore those users of minor platforms,

$$v_e^{(i)} = \arg\max_{C_{\text{min}}^{(i)} \leq v \leq C_{\text{max}}^{(i)}} \frac{P_i(v)}{v}, \tag{21}$$

$$e_t(i) = \frac{P_i(v_e^{(i)})}{v}. \tag{22}$$

$P_i(v)/v$ is the probability of the users' clicks in platform i per unit cost of the advertisement, which represents the efficiency of serving the advertisement to the users of platform i. Higher efficiency of serving the advertisement means less money spent to increase the number of advertisement clicks by one, thus higher promotion of the product with given budget. $v_e^{(i)}$ finds the banner with the highest efficiency, and $e_t(i)$ is the maximal efficiency that can be achieved for platform i. As shown in Fig. 4 (b), the red dashed line is tangent to the curve of $P_i(v) \cdot n_i$, and the tangent point is the banner $v_e^{(i)}$ with the highest efficiency. Apparently, the blue dashed line intersects with the curve at the banner with the highest utility, however, its efficiency is smaller than $v_e^{(i)}$. Combining the efficiency together with the UCB, another mechanism to determine the next platform to serve advertisement is determined as following,

$$\arg\max_i (e_t(i) + b\sqrt{\frac{\ln t}{M_t(i)}}), \tag{23}$$

Since $l_t(i)$ and $e_t(i)$ have different metrics, the user input parameter is set to another positive number b. Eq. (23) can find the platform and the banner with the highest efficiency, providing a feasible way to take advantage of those users in minor platforms which have inadequate users.

The mechanisms based on Eq. (20) and Eq. (23) have different advantages: Eq. (20) finds the platform considering the scales of platforms while Eq. (23) determines the banner based on the efficiency. Only utilizing Eq. (20) may decrease the utility due to waste of budget, while only applying Eq. (23) may lead to the delay even failure of the completion of the advertising within given period. Therefore, the mechanism named Combinatorial Utility and Efficiency Upper Confidence Bound Algorithm (CUE-UCB) is proposed to take advantages of both proposed mechanism.

$$A_t = \begin{cases} \arg\max_i (l_t(i) + a\sqrt{\frac{\ln t}{M_t(i)}}), \text{with Prob. } \alpha; \\ \arg\max_i (e_t(i) + b\sqrt{\frac{\ln t}{M_t(i)}}), \text{with Prob. } 1 - \alpha. \end{cases}$$

α is a user-input parameter determining which metrics is utilized more in the finding of the next banner to place the advertisement. When a new user of platform i arrives, with a probability α, the utility related metrics is chosen to determine the platform A_t to serve the next advertisement. If $i = A_t$, then the user will be advertised using the banner $v_l^{(i)}$; otherwise she will not be advertised. Similarly, the new user of platform i will be considered with probability $1 - \alpha$ using the efficiency related metrics. If $i = A_t$, then the user will be advertised using the banner $v_e^{(i)}$; otherwise she will not be advertised. The proposed CUE-UCB algorithm is summarized in Algorithm 2.

Time complexity. No matter which metrics is chosen, the time complexity to recall function Eq. (18) or Eq. (21) is $\mathcal{O}(m)$, where m is the total number of banners of all k platforms. Eq. (20) and Eq. (23) are both in the order of $\mathcal{O}(k)$, which is also $\mathcal{O}(m)$ since $k \leq m$. The algorithm will

Algorithm 2: Combinatorial Utility and Efficiency Upper Confidence Bound Algorithm (CUE-UCB)

```
Input: Number of MABs $k$, total budget $B$, probability $\alpha$, UCB parameters $a$ and $b$, number of users $n_i$ in each MAB platform $i$, set of banners $\{v_j^{(i)}\}$, set of incoming users $\mathcal{U}$, .
Output: Banner $v^{(i)}$ to serve the advertisement.
$M_t(i) \leftarrow 1, N_j^{(i)} \leftarrow 1, t \leftarrow 1, P_i(v_j^{(i)}) \leftarrow 0, \forall i, j$
for $\forall\ u \in \mathcal{U}$ do
    $f \leftarrow \text{rand}(1)$
    if $f \le \alpha$ then
        for $1 \le i \le k$ do
            $v_l^{(i)} \leftarrow \arg\max_{v_j^{(i)}} \min\{\frac{B}{v_j^{(i)}}, P_i(v_j^{(i)}) \cdot n_i\}$
            $l_t(i) \leftarrow \min\{\frac{B}{v_l^{(i)}}, P_i(v_l^{(i)}) \cdot n_i\}$
        $A_t \leftarrow \arg\max_i (l_t(i) + a\sqrt{\frac{\ln t}{M_t(i)}}), \forall i$
        if $u \in A_t$ then
            Use banner $v_l^{(A_t)}$ to advertise, and collect her response $r_u$,
            $P_i(v_l^{(A_t)}) \leftarrow P_i(v_l^{(A_t)}) + \frac{r_u - P_i(v_l^{(A_t)})}{N_l^{(A_t)}+1}$
            $N_l^{(A_t)} \leftarrow N_l^{(A_t)} + 1, M_t(A_t) \leftarrow M_t(A_t) + 1, t \leftarrow t + 1$
            if $r_u = 1$ then
                $B \leftarrow B - v_l^{(A_t)}$
        else
            Do not serve the advertisement for $u$
    else
        for $1 \le i \le k$ do
            $v_e^{(i)} = \arg\max_{v_j^{(i)}} \frac{P_i(v_j^{(i)})}{v_j^{(i)}}$
            $e_t(i) = \frac{P_i(v_e^{(i)})}{v}$
        $A_t \leftarrow \arg\max_i (e_t(i) + b\sqrt{\frac{\ln t}{M_t(i)}}), \forall i$
        if $u \in A_t$ then
            Use banner $v_e^{(A_t)}$ to advertise, and collect her response $r_u$,
            $P_i(v_e^{(A_t)}) \leftarrow P_i(v_e^{(A_t)}) + \frac{r_u - P_i(v_e^{(A_t)})}{N_e^{(i)}+1}$
            $N_e^{(A_t)} \leftarrow N_e^{(A_t)} + 1, M_t(A_t) \leftarrow M_t(A_t) + 1, t \leftarrow t + 1$
            if $r_u = 1$ then
                $B \leftarrow B - v_e^{(A_t)}$
        else
            Do not serve the advertisement for $u$
```

be executed once for each user, thus $\mathcal{O}(n)$ times for all n users of all the platforms. Therefore, the time complexity of the algorithm is $\mathcal{O}(mn)$.

Definition 6. *Regret. The expected utility of mechanism M is denoted by $U(M, B)$ for a fixed budget B. The expected regret of M is given by $R_M(B) = U(M^*, B) - U(M, B)$, where M^* is the optimal mechanism achieving the maximum utility among all mechanisms. $R_M(B)$ is briefly denoted as $R(B)$ hereafter for conciseness.*

Theorem 4.*The expected regret of mechanism CUE-UCB applied for k-MAB problem is upper-bounded by,*

$$R(B) < \sum_{i:\Delta_i>0} \Delta_i[\sum_{t=1}^{n} \frac{1}{t} \sum_{s=1}^{n} \exp(-\frac{s\epsilon^2}{2}) + 1 + \frac{2}{(\Delta_i - \epsilon)^2}(\log n + \sqrt{\pi \log n} + 1)] + \sum_i (\sum_{j:P_{i,j}<P_i^*} \frac{8\log(B_i/C_{\min}^{(i)})}{\delta_j^{(i)2}} + \frac{\pi^2}{3} + 1) \cdot (P_i^* - P_i(v_j^{(i)})) + \sum_{j:P_{i,j}<P_i^*} \frac{\pi^2(P_{i,j}P_i(v_j^{(i)}) - v^{(i)*}P_i^*)}{6 \cdot v^{(i)*}} + \frac{C_{\min}^{(i)}}{v^{(i)*}}. \tag{24}$$

while the expected average regret w.r.t budget size B approaches to 0 as B goes to infinity.

Proof. The regret of the algorithm mainly comes from two parts. The first part $R_1(B)$ comes from the choosing of sub-optimal MABs due to the use of CUE-UCB algorithm; the second part $R_2(B)$ comes from the rejected offers and wasted budget through overpayment due to the chosen fixed incentive for all the users in the same MAB. $R(B)$ can be denoted as,

$$R(B) = R_1(B) + R_2(B). \tag{25}$$

The regret can be decoupled into these two parts because these two are induced by completely different schemes: the exploration of sub-optimal MABs in the process of finding the optimal MAB, and the fixed price incentivizing mechanism to treat the users as a whole instead of individuals. Meanwhile, these two are also all the reasons contributing to the regret of CUE-UCB apparently. We derive these two regrets separately in the following.

Part I. This part follows the classical proof of UCB related regret. If we treat each MAB as a unique bandit, the proof of UCB of MAB can be utilized to derive the regret where k MABs form a new MAB altogether. W.L.O.G, the 1-st MAB is set to be the optimal MAB with the largest expected reward. The unit regret of choosing suboptimal MAB i is defined as,

$$\Delta_i = \frac{l(1)}{n_1} - \frac{l(i)}{n_i}. \tag{26}$$

The unit regret is the expected difference between the utility increment of the chosen MAB and the optimal MAB for every action. $R_1(B) = \sum_{i:\Delta_i>0} \Delta_i E(T_i(n))$, where $E(T_i(n)) = \sum_{t=1}^{n} \text{II}\{A_t = i\}$ is the expected number of times that i-th MAB is chosen by action A_t after n actions having been taken. Based on the Chernoff-Hoeffding concentration inequality of n independent 1-subgaussian random variables X_t, i.e. $P(\mu \ge \sqrt{\frac{2}{n}\log(\frac{1}{\delta})}) \le \delta$, where $\mu = \sum_{t=1}^{n} X_t/n$, the upper bound of the reward of each MAB can be estimated with good confidence. Then the procedures of estimating the number of times $E(T_i(n))$ that UCB of i-th MAB larger than the 1st MAB, can be utilized as [32], since different MABs can be treated as different independent random variables. According to the results in [32], the regret of the first part is,

$$R_1(B) = \sum_{i:\Delta_i>0} \Delta_i[\sum_{t=1}^{n} \frac{1}{t} \sum_{s=1}^{n} \exp(-\frac{s\epsilon^2}{2}) + 1 + \frac{2}{(\Delta_i - \epsilon)^2} \cdot (\log n + \sqrt{\pi \log n} + 1)], \tag{27}$$

where n is the total number of users across all k MABs, ϵ is a small constant determining the confidence width.

Part II. The regret of the second part is mainly due to rejected offers and the overpayment to the users. Recall that a fixed incentive calculated via CUE-UCB is offered to all the users in the same MAB without differentiating individuals. Therefore, if the incentive is lower than anticipated reward, offers will be rejected; vice versa, if the incentive is higher than anticipation, some budgets are wasted due to overpayment. In [8], a similar fixed price incentivizing mechanism is proposed based on Eq. (18). Since this equation is the only reason causing the regret $R_2(B)$ for CUE-UCB and the same regret in [8], the results in [8] can be extended here. The unit reward of the bandit j in i-th MAB is defined as,

$$F_j^{(i)} = \min\{\frac{B}{v_j^{(i)} \cdot n_i}, P_i(v_j^{(i)})\}, \tag{28}$$

based on which the regret between bandit j and the optimal bandit (denoted by 1 W.L.O.G) for i-th MAB is,

$$\delta_j^{(i)} = F_1^i - F_j^{(i)}. \tag{29}$$

According to Theorem 2 in [8], the upper bound of the regret of this part is equal to,

$$R_2(B) < \sum_i (\sum_{j:P_{i,j}<P_i^*} \frac{8\log(B_i/C^{(i)}_{\min})}{\delta_j^{(i)2}} + \frac{\pi^2}{3} + 1)(P_i^* - P_i(v_j^{(i)})) + \sum_{j:P_{i,j}<P_i^*} \frac{\pi^2(P_{i,j}\cdot P_i(v_j^{(i)}) - v^{(i)*}\cdot P_i^*)}{6\cdot v^{(i)*}} + \frac{C^{(i)}_{\min}}{v^{(i)*}}, \tag{30}$$

where the first and the second term originate from rejected offers and wasted budget through overpayment. P_i^* is the probability of the user acceptance under the optimal incentive in i-th MAB, while $P_{i,j} = P_i(v_j^{(i)})$. The detailed derivation of this regret bound may be referred to [8].

Combining Part I and Part II together gives the upper bound of the regret of the proposed mechanism applied for solving k-MAB, which is,

$$R(B) < \sum_{i:\Delta_i>0} \Delta_i[\sum_{t=1}^{n} \frac{1}{t} \sum_{s=1}^{n} \exp(-\frac{s\epsilon^2}{2}) + 1 + \frac{2}{(\Delta_i-\epsilon)^2}(\log n + \sqrt{\pi \log n} + 1)] + \sum_i (\sum_{j:P_{i,j}<P_i^*} \frac{8\log(B_i/C^{(i)}_{\min})}{\delta_j^{(i)2}} + \frac{\pi^2}{3} + 1)\cdot (P_i^* - P_i(v_j^{(i)})) + \sum_{j:P_{i,j}<P_i^*} \frac{\pi^2(P_{i,j}P_i(v_j^{(i)}) - v^{(i)*}P_i^*)}{6\cdot v^{(i)*}} + \frac{C^{(i)}_{\min}}{v^{(i)*}}. \tag{31}$$

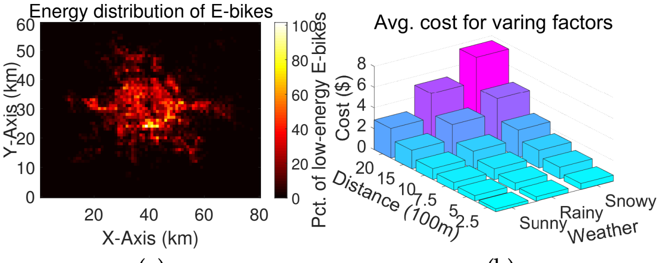
Based on the above upper bound of $R(B)$, the only part influenced by B is $\frac{8\log(B_i/C^{(i)}_{\min})}{\delta_j^{(i)2}}$. Since $B_i \leq B, \forall i$, it can be derived that $R(B)$ is in the order of $\mathcal{O}(\log(B))$. Therefore, the average regret of the proposed mechanism w.r.t budget size B tends to zero, i.e. $\lim_{B\to\infty} \frac{R(B)}{B} = \lim_{B\to\infty} \frac{\mathcal{O}(\log(B))}{B} = 0$.

□

7 Simulation

To evaluate the performance of the proposed mechanisms, we conduct two separate simulations in case of the existing and non-existing of levels of difficulties for k-MAB system. Two different public datasets are utilized in the simulations.

Two applications are necessary here. The first application of sharing E-bikes is the application used to verify our k-level incentivizing mechanism in the sharing economy. Motivated by this work, we further propose k-MAB problem, which requires another more suitable application. Multi-platform advertising is the appropriate application for testing the importance of k-MAB problem since each platform can be treated as one MAB, which forms the k-MAB system altogether.



(a) (b)

Fig. 5: Analysis of dataset and survey (a) distribution of low-energy E-bikes (b) average user cost under various external factors.

	250m	500m	750m	1000m	1500m	2000m
Sunny	0.27	0.51	0.74	1.04	1.77	2.83
Rainy	0.42	0.71	1.05	1.70	3.22	5.19
Snowy	0.58	1.11	1.69	2.61	4.60	7.55

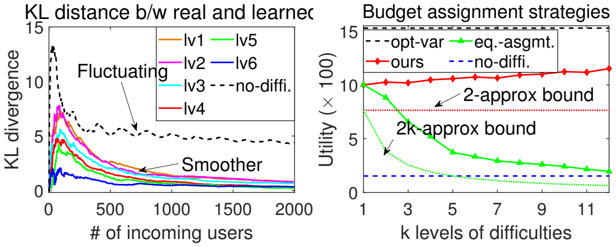
TABLE 1: The average expected cost ($) of users considering different weather conditions and traveling distances.

7.1 Case Study of E-Bike Repositioning

To evaluate the case where k levels of difficulties exist, we conduct a case study based on the popular E-bike sharing system recently[7]. In addition to the re-balancing problem, E-bikes require timely charging for sustainable system utility. The existing solution dispatches maintenance crew to traverse through all energy-demanding stations. To improve efficiency, incentives can be given to users for helping aggregate (low-energy) E-bikes towards some designated stations. The process of determining such incentives directly fits into the framework of the k-level system, where the external factors of weather and (extra) walking distance have impacts on the difficulty of the repositioning tasks. Utility is defined as the number of E-bikes that have been successfully repositioned under a fixed budget.

Due to its nascency and lack of public data for E-bike sharing, we utilize the Mobike dataset[8] instead, assuming the types of bikes have limited impact on the points of interest. The dataset contains 3.2M bicycle trips from May, 10th to 24th in 2017, Beijing, China. Each trip consists of $\langle$*bike type, user id, order id, bike id, starting time, starting location, ending location*$\rangle$. To simulate energy status of E-bikes , we establish an energy model based on the data crawled from XQbike App (E-bike). By tracing each *bike id* with the energy status, locations, the model can closely estimate the residual energy of E-bikes. Note that, this transformation may have limitations as our evaluation includes a subset of all possible routes (without those longer rides using E-bikes). Fig. 5 (a) presents a view of all the energy demand points, with each pixel representing a $100 \times 100m^2$ grid in Beijing. The analysis suggests that if E-bike sharing systems are deployed in such large scale, the maintenance cost is huge with more than 40% E-bikes waiting for recharging.

[7] Bird scooter: https://www.bird.co/
[8] https://biendata.com/competition/mobike



(a) (b)

Fig. 6: Performance comparison (a) learning cost distributions (b) utility and approximation bounds.

To acquire realistic cost distribution of users considering various external factors, we conduct a survey via the Amazon Mechanical Turk (MTurk). The survey starts with introductory questions about the participants' familiarity with the bike-sharing system, and follows by a random combination from different {*weather, walking distance*} to collect the minimum incentives for a repositioning task. A total 385 respondents are received. The average cost is shown in Table 1 and visualized in Fig. 5 (b). It shows that cost in rainy/snowy days are about 2 and 3 times of the cost in sunny days. The cost also grows faster regarding walking distance, which validates that levels of difficulty are indeed heterogenous from the users' perspectives. Based on the surveyed cost distributions, the user cost is randomly sampled from the relevant distribution to simulate the runtime situation in the experiment.

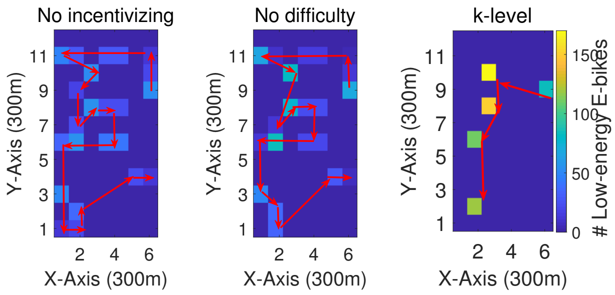
The cost distributions are continuously learned based on users' responses. The closer the learned distributions approximate the private cost from users, the higher chances for the incentivizing offers to get accepted, thus higher overall utility. We use Kullback-Leibler (KL) divergence to measure the difference between the two distributions as used in [22]. Fig. 6 (a) depicts the evolution of KL divergence as the number of users arrive from levels 1-6 vs. the "no-difficulty" approach [9]. It is observed that our mechanism converges much faster and provides a good estimation of the true distribution with 500 users, whereas the no-difficulty approach results 5 times larger KL-divergence with 500 users. The curve also fluctuates due to the dynamic repositioning demands at different stations, which makes it hard to learn a combined distribution. By partitioning tasks into various levels, cost distributions are learned efficiently.

Fig. 6 (b) compares the utility of our mechanism with the *no difficulty* and *equal assignment* mechanisms. The latter assigns equivalent budget to each level. Theoretical bounds from Theorems 1, 2 and the optimal offline solution of OPT-VAR are also plotted. Our mechanism achieves about 7 times utility compared to "no difficulty" and results an actual 1.36 ratio to OPT-VAR in the evaluation. Introducing more levels, the utility climbs up since our mechanism could adaptively assign budgets among all the levels. In contrast, "equal assignment" trends down since the difficult levels demand more budgets while the easy levels have surplus, thereby leading to inefficient use of the budget among different levels. It converges to the bottom line of "no difficulty" when more levels are treated equally. The result validates substantial improvement of utility by considering task difficulty as context information and highlights the importance of our mechanism that utilizes the budget efficiently across all levels.

Table 2 further evaluates the budget and time required to reach a utility objective of 1500 tasks. The average incentive for each user is 1.16$ in our mechanism, 1.9 times of the optimal solution; whereas no difficulty provides 2.96$, 5 times of the optimal. Our mechanism also saves 54% time to accomplish the utility objective much faster since the provided incentives can reflect the true cost of users much better, thereby receiving less "decline".

	no-diffi.	equal asgmt.	ours	OPT-VAR*
budget ($)	4438	3729	**1733**	890*
incent. ($)	2.96	2.49	**1.16**	0.59*
time (min)	325	177	**151**	67*

TABLE 2: Budget and time required to achieve the utility objective.



(a) (b) (c)

Fig. 7: Maintenance overhead in E-bike reposition (a) no incentive (b) no difficulty (c) k-level difficulty.

Fig. 7 shows a running example if the maintenance crew visits the stations where the E-bikes are aggregated by the incentivizing mechanisms. "No difficulty" mainly repositions E-bikes in close distance, but fails to look further. Our mechanism surpasses no difficulty by aggregating the low-energy E-bikes at fewer stations. The maintenance crew would travel 10.9, 9.5, and 4.3 km accordingly, with a sheer 55% and 61% mileage saving regarding no difficulty and no incentive approaches.

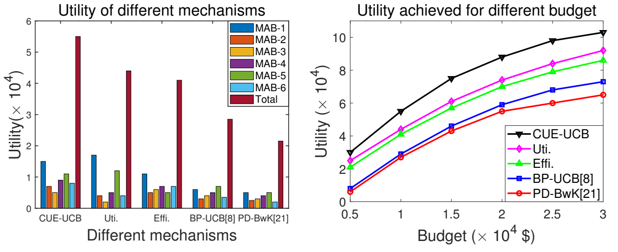
7.2 Case Study of Mobile Advertisements Click

To evaluate the performance of the CUE-UCB mechanism for the generalized k-MAB system, we utilize the public dataset of clicks on the mobile advertisements collected by Avazu [9]. This dataset contains 40M users' traces of click on mobile advertisements from Oct. 21st to 30th in 2014. 4M traces are randomly sampled from the dataset to efficiently reduce the scale of the data while also represent the typical trends. Each trace consists of ⟨*click, hour, site_id, app_id, device_ip, anonymized features*⟩ etc. To accommodate to the scenario of k-MAB, each APP identified by the *app_id* is considered as one MAB, each site in the APP identified by the *site_id* is the banner, and each user is identified using *device_ip*. However, the Avazu dataset does not include the cost of serving the advertisement in each banner, so we use synthetic data to generate the cost. In the reality, platforms with higher volume of the flow of users usually charge more for the advertising, i.e. the cost per advertisement is positively related to the amount of users. However, if the cost is set to be proportional to the user amount, platform A charges 10 times compared with platform B is apparently unreasonable. Therefore, the synthetic cost of each banner based on the natural logarithm is generated as following,

$$v_j^{(i)} = (\ln(\text{NUM\_APP}) + \ln(\text{NUM\_SITE}) + \delta_j)/d, \quad (32)$$

where NUM_APP and NUM_SITE are the total number of the appearance of APP (MAB) i and SITE (banner) j, δ_j is a randomly generated number, d is an adjustable positive

[9] https://www.kaggle.com/c/avazu-ctr-prediction/data



(a)

(b)

Fig. 8: Utility of different mechanisms (a) for the same budget (b) for different budgets.

	CUE-UCB	Uti.	Effi.	BP-UCB	PD-BwK
KL-div. ($)	**0.08**	0.10	0.13	0.19	0.23
time (M)	0.63	0.59	0.69	**0.38**	0.47

TABLE 3: Budget and time required to achieve the utility objective for different mechanisms (BP-UCB [8] and PD-BwK [23] are benchmarks).

number. We summarize the number of appearance of all APP and sites, and pick the top 10 APPs according to the number as the 10 MABs (i.e. $k = 10$). We also calculate the Click-Through-Rate (CTR) for each site, which is the number of clicks per impression, where the highest CTR is found to be about 0.45. The found CTR is used as the intrinsic probability that the user will click on the advertisement, based on which a statical model is generated to simulate the user's behavior.

When a user arrives, CUE-UCB algorithm determines the best banner to place the advertisement, and the advertisement will be served if the user is using that APP, and her response is collected. α, a, b, d are set to be $0.2, 1000, 10, 50$, and the budget is set to be 10000 \$. The probability α value achieving the highest utility is found in the range of $[0, 1]$ with an increment of 0.05. In order to make $l_t(i)$ and $e_t(i)$ comparable to their UCBs in scale, the a and b values are chosen in the range of $[500, 5000]$ and $[10, 100]$ (ranges determined by the ratios between the utility and UCB when $t = 10^3$ and $t = 10^6$) with an increment of 500 and 10, which achieves the highest utility with other parameters fixed. d is chosen to make the average incentive provided in the simulation to be about 0.1 \$ per user, referring to the advertising cost on Youtube per click [37].

As shown in Fig. 8 (a), the utility (number of clicks of advertisements) achieved using different algorithms is compared between different algorithms. Our CUE-UCB algorithm is compared with the benchmark algorithms BP-UCB proposed in [8] and PD-BwK proposed in [23], the "Uti." only using utility function, and the "Effi." only using the efficiency algorithm. Note that BP-UCB and PD-BwK can just handle the case of one MAB, which are adjusted to treat all banners in different platforms as a whole in the simulation. The simulation shows that CUE-UCB achieve about 5.5×10^4 utility, which improves the utility by about 90% and 150% compared with the BP-UCB and PD-BwK. Our algorithm treats different platforms separately, and always serve the advertisement to the best banner in the best platform, which avoids the waste of budget compared with them. Meanwhile, only using the utility function or efficiency function achieves lower utility since they either choose many inefficient users or the scale of the chosen platform is not enough.

Fig. 8 (b) compares the utility achieved for different budgets varying from 5000\$ to 30000\$. It can be observed that, the increment of utility decreases as the budget increases, since most of high efficient users have been served with the advertisement, and the extra budget has to be used to advertise less efficient users. The difference between CUE-UCB and the benchmarks increases as the budget increases, however, the percentage of the improvement decreases accordingly. The highest percentages of the improvement of the utility are about 275% and 370% when budget=5000 \$, while the same percentages decrease to 41% and 59 % when budget=30000 \$, since the advantages of our mechanism is diminished when the system has to choose more less-efficient users.

We also evaluate the precision of the learned probability of each banner using KL-divergence, and also the time needed to complete the advertising. The proposed CUE-UCB mechanism achieves the minimal KL-divergence, which is only about 42% and 34% of two benchmarks. However, our mechanism needs about 65% more time to finish the advertising since it may skip some incoming users and choose to serve the advertisement for some more efficient users. We believe the tradeoff of time is worthwhile since the utility can increase up to 275% when the time constraint is relaxed.

Note that, most of the mechanisms solving MAB can not be used as benchmarks here since they may not be able to address the case of MAB with Knapsacks nor to form a fixed price incentivizing mechanism, which are both basic requirements of this paper. Therefore, BP-UCB and PD-BwK are utilized as two competitive state of the art benchmarks.

8 Conclusion

In this paper, we propose a new reinforcement learning problem named k-MAB, and study the mechanism to address the problem with and without the contextual information of difficulties. Given levels of difficulties, we partition the tasks into heterogeneous levels of difficulty based on the external factors that may impact on users' cost. We formally analyze the ratios between assigning varied and fixed incentives in different scenarios and design a mechanism to learn users' cost distributions via minimizing the regret. For more generalized k-MAB problem, we propose CUE-UCB algorithm to jointly consider the utility function and the efficiency of serving the advertisements. We present two case study based on the E-bike sharing system and mobile advertisements click using public datasets. The results demonstrate dramatic improvement in utility and learning by utilizing our mechanism compared with benchmarks.

9 Acknowledgment

The work in this paper was supported by the U.S. National Science Foundation under grant number CNS-1513719 and award number 2044841.

References

[1] Y. Li, Y. Zheng, and Q. Yang, "Dynamic bike reposition: A spatio-temporal reinforcement learning approach", *In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (SIGKDD'18)*. 1724-1733.

[2] J. Spero, "Ofo shuts international division as staff prepare for bankruptcy", 2019. Retrived from https://www.ft.com/content/e23a3480-141b-11e9-a581-4ff78404524e.

[3] X. Zhang, Z. Yang, Z. Zhou, H. Cai, L. Chen, and X. Li, "Free market of crowdsourcing: Incentive mechanism design for mobile sensing", *IEEE Trans. Parallel. Distrib. Syst.* 25, 12 (2014), 3190-3200.

[4] R. Zhou, Z. Li, and C. Wu, "A truthful online mechanism for location-aware tasks in mobile crowd sensing", *IEEE Trans. Mob. Comput.* 17, 8 (2017), 1737-1749.

[5] P. Zhou, C. Wang, and Y. Yang, "Self-sustainable sensor networks with multi-source energy harvesting and wireless charging", *In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'19)*. 1828-1836.

[6] A. Archer, and E. Tardos, "Frugal path mechanisms", *ACM Trans. Algorithms* 3, 1 (2007), 1-22.

[7] Y. Singer, "Budget feasible mechanisms", *In Proceedings of the 51st IEEE Annual Symposium on Foundations of Computer Science (FOCS'10)*. 765-774.

[8] A. Singla, and A. Krause, "Truthful incentives in crowdsourcing tasks using regret minimization mechanisms", *In Proceedings of the 22nd International Conference on World Wide Web (WWW'13)*. 1167-1178.

[9] A. Singla, M. Santoni, G. Bartok, P. Mukerji, M. Meenen, and A. Krause, "Incentivizing users for balancing bike sharing systems", *In Proceedings of the 29th AAAI conference on Artificial Intelligence (AAAI'15)*. 723-729.

[10] P. Zhou, C. Wang, Y. Yang, X. Wei, "E-Sharing: Data-driven Online Optimization of Parking Location Placement for Dockless Electric Bike Sharing", *In Proceedings of the 40th IEEE International Conference on Distributed Computing Systems (ICDCS'20)*. 474-484.

[11] C. Hirnschall, A. Singla, S. Tschiatschek, and A. Krause, "Learning user preferences to incentivize exploration in the sharing economy", *In Proceedings of the 32th AAAI conference on Artificial Intelligence (AAAI'18)*. 2248-2256.

[12] C. Ho, and J. W. Vaughan, "Online task assignment in crowdsourcing markets", *In Proceedings of the 26th AAAI conference on Artificial Intelligence (AAAI'12)*. 45-51.

[13] G. Goel, A. Nikzad, and A. Singla, "Mechanism design for crowdsourcing markets with heterogeneous tasks", *In Proceedings of 2nd AAAI Conference on Human Computation and Crowdsourcing (HCOMP'14)*. 77-86.

[14] P. Zhou, X. Wei, C. Wang, and Y. Yang, "Explore truthful incentives for tasks with heterogenous levels of difficulty in the sharing economy", *In Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*. 665-671.

[15] S. Assadi, J. Hsu, and S. Jabbari, "Online assignment of heterogeneous tasks in crowdsourcing markets", *In Proceedings of 3rd AAAI Conference on Human Computation and Crowdsourcing (HCOMP'15)*. 12-21.

[16] W. H. Press, "Bandit solutions provide unified ethical models for randomized clinical trials and comparative effectiveness research", *In Proceedings of the National Academy of Sciences* 106, 52 (2009), 22387-22392.

[17] R. B. Myerson, "Optimal auction design", *Math. Oper. Res.* 6, 1 (1981), 58-73.

[18] A. Badanidiyuru, R. Kleinberg, and Y. Singer, "Learning on a budget: posted price mechanisms for online procurement", *In Proceedings of the 13th ACM Conference on Electronic Commerce (EC'12)*. 128-145.

[19] S. Dobzinski, N. Nisan, and M. Schapira, "Truthful randomized mechanisms for combinatorial auctions", *In Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC'06)*. 644-652.

[20] R. Kleinberg, A. Niculescu-Mizil, and Y. Sharma, "Regret bounds for sleeping experts and bandits", *Mach. Learn.* 80, 2 (2010), 245-272.

[21] A. A. Deshmukh, S. Sharma, J. W. Cutler, M. Moldwin, and C. Scott, "Simple regret minimization for contextual bandits", *arXiv preprint*. arXiv:1810.07371 (2018).

[22] Z. Liu, Y. Shen, and Y. Zhu, "Inferring dockless shared bike distribution in new cities", *In Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM'18)*. 378-386.

[23] A. Badanidiyuru, R. Kleinberg, and A. Slivkins, "Bandits with knapsacks", *In the 54th IEEE Annual Symposium on Foundations of Computer Science (FOCS'13)*. 207-216.

[24] F. Li, J. Liu, and B. Ji, "Combinatorial Sleeping Bandits with Fairness Constraints", *IEEE Trans. Netw. Sci. Eng.* 7, 3 (2019), 1799-1813.

[25] Y. Liu, and C. Ho, "Incentivizing high quality user contributions: New arm generation in bandit learning", *In Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI'18)*. 1146-1153.

[26] L. Chen, K. Cai, L. Huang, and J. Lui, "Beyond the Click-Through Rate: Web Link Selection with Multi-level Feedback", *In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*. 3308-3314.

[27] A. Badanidiyuru, J. Langford, and A. Slivkins, "Resourceful contextual bandits", *In Proceedings of Conference on Learning Theory (PMLR'14)*. 1109-1134.

[28] Y. Yue, J. Broder, R. Kleinberg, and T. Joachims, "The k-armed dueling bandits problem", *J. Comput. Syst. Sci.* 78, 5 (2012), 1538-1556.

[29] R. Combes, Richard, M.S.T.M Shahi, and A. Proutiere, "Combinatorial bandits revisited", *In Proceedings of the 29th Conference on Neural Information Processing Systems NeurIPS'15*.

[30] "How Much Does YouTube Advertising Cost?", *WebFX*, 2019. Retrived from https://www.webfx.com/internet-marketing/how-much-does-youtube-advertising-cost.html.

[31] A. Agarwal, H. Luo, B. Neyshabur, and R. E. Schapire," Corralling a band of bandit algorithms", *In Proceedings of Conference on Learning Theory (PMLR'17)*. 12-38.

[32] T. L. Lai, H. Robbins, "Asymptotically efficient adaptive allocation rules", *Adv. Appl. Math. Mech.* 6, 1 (1985), 4-22.

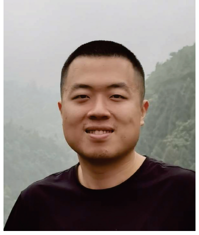
[33] L. Pan, Q. Cai, Z. Fang, P. Tang, and L. Huang, "A deep reinforcement learning framework for rebalancing dockless bike sharing systems", *In Proceedings of the 33th AAAI Conference on Artificial Intelligence (AAAI'19)*. 1393-1400.

[34] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning", *IEEE Internet Things J.* 7, 7 (2020), 6360-6368.

[35] T. Lu, D. Pal, and M. Pal, "Contextual multi-armed bandits", In *In Proceedings of the 30th International Conference on Artificial Intelligence and Statistics (AISTATS'10)*. 485-492.

[36] M. Collier, and H. U. Llorens, "Deep contextual multi-armed bandits", *arXiv preprint*. arXiv:1807.09809 (2018).

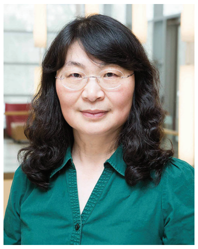
[37] "How much does Youtube Advertising Cost?", *WebFX*, 2021. Retrived from https://www.webfx.com/internet-marketing/how-much-does-youtube-advertising-cost.html.



Pengzhan Zhou received the B.S. degree in both Applied Physics and Applied Mathematics from Shanghai Jiaotong University, Shanghai, China. He received Ph.D. degree in Computer and Electrical Engineering from Stony Brook University, NY, USA. His research interests include machine learning, wireless sensor networks, performance evaluation of network protocols and algorithms.

Xin Wei received the bachelor degree in Economics from Shanghai Jiaotong University, China. She is pursuing the Ph.D. degree at the Department of Computer Science at Old Dominion University, Norfolk, VA. Her research interest mainly lies in machine learning.

Cong Wang received the B. Eng degree in Information Engineering from the Chinese University of Hong Kong in 2008, M.S. degree in Electrical Engineering from Columbia University in 2009, and Ph.D. in Computer and Electrical Engineering from at Stony Brook University, NY, in 2016. He is currently an Assistant Professor at the Department of Cybersecurity, George Mason University, Fairfax, VA. His research focuses on addressing security and privacy challenges in Mobile, Cloud Computing, IoT, Machine Learning and System. He is the recipient of IEEE PERCOM Mark Weiser Best Paper Award in 2018, Commonwealth Cyber Initiative Research and Innovation Award, and NSF CAREER Award in 2021.



Yuanyuan Yang received the BEng and MS degrees in computer science and engineering from Tsinghua University, Beijing, China, and the MSE and PhD degrees in computer science from Johns Hopkins University, Baltimore, Maryland. She is a SUNY Distinguished Professor of computer engineering and computer science at Stony Brook University, New York, and is currently on leave at the National Science Foundation as a Program Director. Her research interests include edge computing, data center networks, cloud computing and wireless networks. She has published over 400 papers in major journals and refereed conference proceedings and holds seven US patents in these areas. She is currently the Associate Editor-in-Chief for IEEE Transactions on Cloud Computing and an Associate Editor for ACM Computing Surveys. She has served as an Associate Editor-in-Chief and Associated Editor for IEEE Transactions on Computers and Associate Editor for IEEE Transactions on Parallel and Distributed Systems. She has also served as a general chair, program chair, or vice chair for several major conferences and a program committee member for numerous conferences. She is an IEEE Fellow.