Explore Truthful Incentives for Tasks with Heterogenous Levels of Difficulty in the Sharing Economy

Pengzhan Zhou[1] , **Xin Wei**[2] , **Cong Wang**[2] and **Yuanyuan Yang**[1]
[1]Dept. of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA
[2]Dept. of Computer Science, Old Dominion University, Norfolk, VA 23529, USA
{pengzhan.zhou, yuanyuan.yang}@stonybrook.edu, {xwei001, c1wang}@odu.edu

Abstract

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 general 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. Our experiment demonstrates utility improvement about 7 times and time saving of 54% to meet a utility objective compared to the previous works.

1 Introduction

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 [Li *et al.*, 2018] or send maintenance crew for charging the e-scooters. Such large maintenance overhead drives several bike-sharing platforms to the verge of bankruptcy recently [Spero, 2019].

Previous research proposed to seek user cooperation with monetary incentives. Incentives are provided in mobile sensing tasks [Zhang *et al.*, 2014; Zhou *et al.*, 2018; 2019], 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 [Archer and Tardos, 2007; Singer, 2010], incentives are explored in crowdsourcing tasks to learn private cost distribution and maximize utility [Singla and Krause, 2013]. 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 [Singla *et al.*, 2015]. Similarly, incentives are offered to encourage users for taking different options such as renting an apartment with no review rating [Hirnschall *et al.*, 2018].

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 [Singla and Krause, 2013] 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 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. Finally, we conduct a case study of electric bike-sharing and evaluate the proposed mechanism on a public dataset. Compared to the previous works, the experiments demonstrate that our mechanism not only achieves about 7 times utility, but also saves 54% time to reach a utility objective.

2 Preliminary

2.1 Motivation

The previous works explore the distribution of user cost to find the optimal incentivizing strategies. Nevertheless, they assume each user 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) or new review ratings (housing rental). These factors could cause users' cost to vary, depending on how users perceive the task at a different time. The cost of users 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.

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*}[1] After returning the bikes to a different station, the user may have to walk extra distance to her destination. In [Singla *et al.*, 2015], 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 [Ho and Vaughan, 2012; Goel *et al.*, 2014; Assadi *et al.*, 2015]. 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.

[1] 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.

2.2 System Model

Definition 1. *Task difficulty. Each level of difficulty is defined by a point in the space of external factors.*

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 [Myerson, 1981; Badanidiyuru *et al.*, 2012]. Instead of building on truthful bidding/auction mechanisms such as second-price auction [Dobzinski *et al.*, 2006], 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 2. *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 2*, 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,

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

Definition 3. *Utility. The number of tasks completed by the users via the incentivizing mechanism.*

Definition 4. *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.

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

Definition 6. *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 $C^{(i)}_{q_i}$ to the first q_i users in the i-th level, where q_i is the largest number such that $C^{(i)}_{q_i} \le \frac{B_i}{q_i}$, $\forall 1 \le i \le k$.*

Proof. For any $1 \le j \le q_i$, since $C^{(i)}_j \le C^{(i)}_{q_i}$, providing $C^{(i)}_{q_i}$ ensures that the first q_i users would accept the task. Meanwhile, $C^{(i)}_{q_i} \cdot q_i \le B_i$ 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 $C^{(i)}_{q'_i} \cdot q'_i \le B_i$. However, q_i is the largest number satisfying $C^{(i)}_{q_i} \le \frac{B_i}{q_i}$ for budget feasibility, thereby causing a contradiction. The lemma is proved. □

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

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

Proof. We prove this theorem by mathematical induction.
Base case: The work of [Singer, 2010] has proved this base case when $k = 1$ (only one level of difficulty).
Inductive step: For $k \ge 1$, assume that $l^*(k) \le 2k \cdot l(k)$ holds, we want to prove that $l^*(k+1) \le 2(k+1) \cdot l(k+1)$ 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 $l^*(k+1) \le 2(k+1) \cdot l(k+1)$, it is sufficient to prove that both 1): $l^*(k) \le 2k \cdot l(k+1)$ and 2) $\Delta l^* \le 2 \cdot l(k+1)$ hold.

1) We prove $l^*(k) \le 2k \cdot l(k+1)$. 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). Hence, the utility of $(k+1)$-level system is at least as good as the k level: $l(k) \le l(k+1)$. Plug in this into baseline assumption $l^*(k) \le 2k \cdot l(k)$, then $l^*(k) \le 2k \cdot l(k+1)$ is proved.

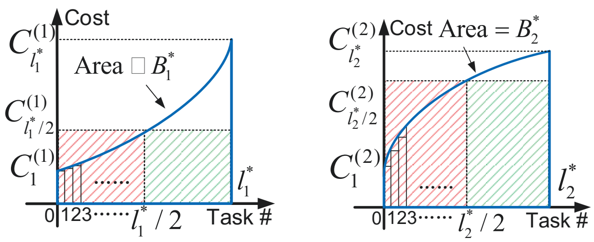
2) We prove $\Delta l^* \le 2 \cdot l(k+1)$. The sketch is to apply Eq. (1), which implies that the cost in the $(k+1)$-th level is larger than the k-th level for the same position j in the sorted list. Since the cost is relatively higher at $(k+1)$-th level, the number of tasks that can be successfully performed is no greater than that from level k, i.e. $l^*_{k+1}(k+1) \le l^*_k(k)$. Similarly, for k levels, $l^*_k(k) \le l^*_i(k)$ and $l^*(k) = \sum_{i=1}^{k} l^*_i(k)$, from which it can be derived that $l^*_k(k) \le \frac{l^*(k)}{k}$. 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,

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

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, $C^{(1)}_1 \le C^{(1)}_2 \le \ldots \le C^{(1)}_{n_1}$ and $C^{(2)}_1 \le C^{(2)}_2 \le \ldots \le C^{(2)}_{n_2}$. 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 ($l^* = l^*_1 + l^*_2$)[3]. 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.

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



(a) (b)

Figure 1: 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^*.

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. 1, 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).

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. 1 (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*,

$$B_i \geq \frac{l_i^*}{2} \cdot C^{(i)}_{l_i^*/2} \Longrightarrow l_i \geq \frac{l_i^*}{2}. \tag{3}$$

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

$$l_i \geq \frac{l_i^*}{2} \Longrightarrow \sum_{i=1}^{k} l_i \geq \sum_{i=1}^{k} \frac{l_i^*}{2} \Longrightarrow l \geq \frac{l^*}{2}. \tag{4}$$

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. 1 still hold for any level i of the k-level system,

$$\frac{l_i^*}{2} \cdot C^{(i)}_{l_i^*/2} \leq B_i^* \Longrightarrow B_i \leq B_i^* \Longrightarrow \sum_{i=1}^{k} B_i \leq B. \tag{5}$$

□

4 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 [Kleinberg *et al.*, 2010; Deshmukh *et al.*, 2018]. 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,

$$v^{(i)} = \underset{C^{(i)}_{\min} \leq v \leq C^{(i)}_{\max}}{\arg\max} \min\left\{\frac{B_i}{v}, P_i(v) \cdot n_i\right\}. \tag{6}$$

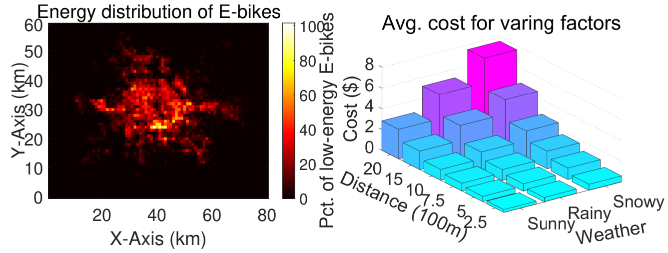
v is a discrete variable in the range of $C^{(i)}_{\min}$ and $C^{(i)}_{\max}$, which are the minimum and maximum incentives allowed in level i. $\frac{B_i}{v}$ is the number of tasks that can be completed with budget B_i by running with incentive v. n_i is the number of users performing level i tasks. $P_i(v)$ 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

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_min^(i) and C_max^(i) for level i, incentive increment Δv, set of incoming users U.
Output: Incentive v^(i) for incoming users at level i.
S ← 0, S_i ← 0, N_j^(i) ← 0, l ← 0, l_i ← 0, ∀i, j
for ∀ u ∈ U do
    Determine the level i that u belongs to
    v_j^(i) ← C_min^(i) + (j − 1) · Δv, ∀j
    v^(i) ← argmax_{C_min^(i) ≤ v_j^(i) ≤ C_max^(i)} min{ B_i / v_j^(i), P_i(v_j^(i)) · n_i }
    if S_i + v^(i) ≤ B_i then
        Provide v^(i) to u, and collect her response r_u
        P_i(v^(i)) = P_i(v^(i)) + (r_u − P_i(v^(i))) / (N_j^(i) + 1),
        N_j^(i) ← N_j^(i) + 1
        l ← l + r_u, l_i ← l_i + r_u //Update utility
        if r_u = 1 then
            S_i ← S_i + v^(i) //The total used budget
    n_j^(i) ← n_i(P_i(v_j^(i)) − P_i(v_{j−1}^(i))), ∀ i, j
    Sort v_j^(i) in an ascending order, getting sequence V
    while S < B, & V ≠ φ do
        Extract level i and order j from V_1 //Finding l_i^*
        if S + n_j^(i) v_j^(i) < B then
            S ← S + n_j^(i) v_j^(i), l_i^* ← l_i^* + n_j^(i), V ← V \ V_1
    B_i ← (l_i^*/2) · C_{l_i^*/2}^(i), ∀i //According to Theorem 1
    β ← B / Σ_{i=1}^k B_i; B_i ← B_i · β, ∀i = 1, 2, ..., k
    U ← U \ u //Remove u, and process the next user
```

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.

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.



(a) (b)

Figure 2: 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.

5 Case Study of E-Bike Repositioning

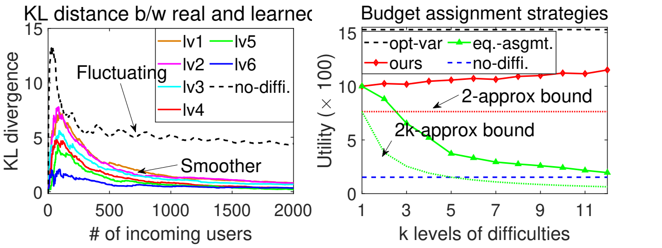
To evaluate performance, we conduct a case study based on the popular E-bike sharing system recently[4]. In addition to the re-balancing problem, E-bikes require timely charging for sustainable system utility. The existing solution dispatches maintenance crew to traverse through all energy-demanding 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[5] 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 ⟨*bike type, user id, order id, bike id, starting time, starting location, ending location*⟩. 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. 2 (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.

To acquire realistic cost distribution of users considering various external factors, we conduct a survey via the Ama-

[4]Bird scooter: https://www.bird.co/

[5]https://biendata.com/competition/mobike



(a) (b)

Figure 3: Performance comparison (a) learning cost distributions (b) utility and approximation bounds.

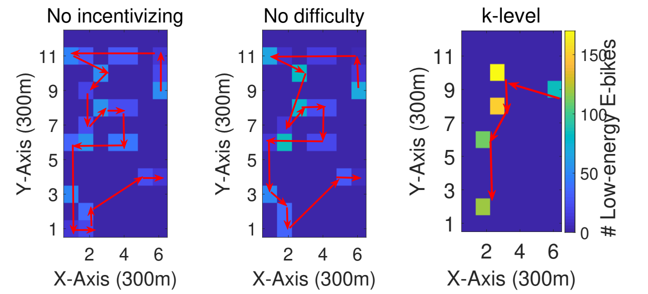
zon Mechanical Turk (MTurk). The survey starts with introductory questions about the participants' familiarity with the bike-sharing system, and follows by a random combination from different {*weather, walking distance*} to collect the minimum incentives for a repositioning task. A total 385 respondents are received. The average cost is shown in Table 1 and visualized in Fig. 2 (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 run-time 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 [Liu *et al.*, 2018]. Fig. 3 (a) depicts the evolution of KL divergence as the number of users arrive from levels 1-6 vs. the "no-difficulty" approach [Singla *et al.*, 2015]. 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. 3 (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.

	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)

Figure 4: Maintenance overhead in E-bike reposition (a) no incentive (b) no difficulty (c) k-level difficulty.

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".

Fig. 4 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.

6 Conclusion

In this paper, 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. We present a case study based on the E-bike sharing system using public dataset. The results demonstrate dramatic improvement in utility and learning by considering the levels of task difficulty.

Acknowledgments

This work was supported in part by the U.S. National Science Foundation under grant numbers CNS-1513719 and CCF-1850045.

References

[Archer and Tardos, 2007] Aaron Archer and Éva Tardos. Frugal path mechanisms. *ACM Transactions on Algorithms (TALG)*, 3(1):3, 2007.

[Assadi *et al.*, 2015] Sepehr Assadi, Justin Hsu, and Shahin Jabbari. Online assignment of heterogeneous tasks in crowdsourcing markets. In *3rd AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, 2015.

[Badanidiyuru *et al.*, 2012] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Yaron Singer. Learning on a budget: posted price mechanisms for online procurement. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC)*, pages 128–145, 2012.

[Deshmukh *et al.*, 2018] Aniket Anand Deshmukh, Srinagesh Sharma, James W Cutler, Mark Moldwin, and Clayton Scott. Simple regret minimization for contextual bandits. *arXiv preprint arXiv:1810.07371*, 2018.

[Dobzinski *et al.*, 2006] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Truthful randomized mechanisms for combinatorial auctions. In *Proceedings of the 38th annual ACM Symposium on Theory of Computing (STOC)*, pages 644–652, 2006.

[Goel *et al.*, 2014] Gagan Goel, Afshin Nikzad, and Adish Singla. Mechanism design for crowdsourcing markets with heterogeneous tasks. In *2nd AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, 2014.

[Hirnschall *et al.*, 2018] Christoph Hirnschall, Adish Singla, Sebastian Tschiatschek, and Andreas Krause. Learning user preferences to incentivize exploration in the sharing economy. In *32nd AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[Ho and Vaughan, 2012] Chien-Ju Ho and Jennifer Wortman Vaughan. Online task assignment in crowdsourcing markets. In *26th AAAI Conference on Artificial Intelligence (AAAI)*, 2012.

[Kleinberg *et al.*, 2010] Robert Kleinberg, Alexandru Niculescu-Mizil, and Yogeshwer Sharma. Regret bounds for sleeping experts and bandits. *Machine Learning*, 80(2-3):245–272, 2010.

[Li *et al.*, 2018] Yexin Li, Yu Zheng, and Qiang Yang. Dynamic bike reposition: A spatio-temporal reinforcement learning approach. In *Proceedings of the 24th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 1724–1733, 2018.

[Liu *et al.*, 2018] Zhaoyang Liu, Yanyan Shen, and Yanmin Zhu. Inferring dockless shared bike distribution in new cities. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 378–386, 2018.

[Myerson, 1981] Roger B Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[Singer, 2010] Yaron Singer. Budget feasible mechanisms. In *IEEE 51st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 765–774, 2010.

[Singla and Krause, 2013] Adish Singla and Andreas Krause. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *Proceedings of the 22nd International Conference on World Wide Web (WWW)*, pages 1167–1178, 2013.

[Singla *et al.*, 2015] Adish Singla, Marco Santoni, Gábor Bartók, Pratik Mukerji, Moritz Meenen, and Andreas Krause. Incentivizing users for balancing bike sharing systems. In *29th AAAI Conference on Artificial Intelligence (AAAI)*, pages 723–729, 2015.

[Spero, 2019] Josh Spero. Ofo shuts international division as staff prepare for bankruptcy, 2019.

[Zhang *et al.*, 2014] Xinglin Zhang, Zheng Yang, Zimu Zhou, Haibin Cai, Lei Chen, and Xiangyang Li. Free market of crowdsourcing: Incentive mechanism design for mobile sensing. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 25(12):3190–3200, 2014.

[Zhou *et al.*, 2018] Ruiting Zhou, Zongpeng Li, and Chuan Wu. A truthful online mechanism for location-aware tasks in mobile crowd sensing. *IEEE Transactions on Mobile Computing (TMC)*, 17(8):1737–1749, 2018.

[Zhou *et al.*, 2019] Pengzhan Zhou, Cong Wang, and Yuanyuan Yang. Self-sustainable sensor networks with multi-source energy harvesting and wireless charging. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2019.