Mechanism Design with Predicted Task Revenue for Bike Sharing Systems

Hongtao Lv,[1] **Chaoli Zhang,**[1] **Zhenzhe Zheng,**[1] **Tie Luo,**[2] **Fan Wu,**[1*] **Guihai Chen**[1]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
[2] Department of Computer Science, Missouri University of Science and Technology, Rolla, USA
{lvhongtao, chaoli_zhang, zhengzhenzhe}@sjtu.edu.cn, tluo@mst.edu, fwu@sjtu.edu.cn, gchen@cs.sjtu.edu.cn

Abstract

Bike sharing systems have been widely deployed around the world in recent years. A core problem in such systems is to reposition the bikes so that the distribution of bike supply is reshaped to better match the dynamic bike demand. When the bike-sharing company or platform is able to predict the revenue of each reposition task based on historic data, an additional constraint is to cap the payment for each task below its predicted revenue. In this paper, we propose an incentive mechanism called FEITE to incentivize users to park bicycles at locations desired by the platform toward rebalancing supply and demand. FEITE possesses four important economic and computational properties such as truthfulness and budget feasibility. Furthermore, we prove that when the payment budget is tight, the overall revenue will still exceed or equal the budget. Otherwise, FEITE achieves 2-approximation as compared to the optimal (revenue-maximizing) solution, which is close to the lower bound of at least $\sqrt{2}$ that we also prove. Using an industrial dataset obtained from a large bike-sharing company, our experiments show that FEITE is effective in rebalancing bike supply and demand and generating high revenue as a result, which outperforms several benchmark mechanisms.

Introduction

Bike sharing is a new transportation mode with many benefits in offering convenience and flexibility as well as lowering economic cost. By 2015, more than 7000 bike sharing systems have been deployed around the world (Laporte, Meunier, and Calvo 2015). However, the flexibility of bike sharing systems, in particular the "anywhere-parking" convenience, brings forth a serious issue of imbalance between the distribution of bike supply and demand. This leads to many users being unable to find a bicycle nearby when they need it, and ultimately affects company revenue adversely. Hence there is an urgent need to rebalance the supply and demand by repositioning the bicycles, which we refer to as a bike rebalancing problem.

There are two approaches to solving this problem. One is to relocate bicycles by the staff of the bike-sharing com-

pany or platform, for example using trucks. This involves route planning and typically uses linear programming techniques, as has been extensively studied with static models (Maggioni et al. 2019; Schuijbroek, Hampshire, and Van Hoeve 2017) and dynamic models (Kek et al. 2009; Angeloudis, Hu, and Bell 2014). However, this approach is costly and not eco-friendly in terms of carbon footprint.

Another approach is to design incentive mechanisms to motivate users to reposition bicycles at platform-desired locations for rebalancing supply and demand. This falls under the research area of crowdsourcing (Fricker and Gast 2016) which takes advantage of the power of crowd to complete tasks that are otherwise difficult (Luo et al. 2016). With this idea, Ghosh and Varakantham (2017) proposed a solution that generates repositioning tasks together with the use of bike trailers, and pays users using the Vickrey-Clarke-Groves (VCG) mechanism. Singla et al. (2015) introduced a crowdsourcing method that offers users monetary incentive for parking their bikes at recommended locations.

However, most of existing mechanisms for this problem have not considered the predicted revenue (or value) of a repositioning task. Such predicted values are made available in recent years, due to the prosperity of deep learning technique which can well predict the bike demand in such systems (Yang et al. 2016; Li and Zheng 2019; Zhang et al. 2016). With this predictive power, the expected revenue or value of a repositioning task can be easily obtained. Therefore, the platform would assign a task to a user only if the payment to the user does not exceed the predicted revenue. This constraint is akin to the "reserve price" in forward auctions which are used to promote revenue. In our problem, it is used to better control the payments and thereby promote the profit of the platform. This constraint is generally overlooked in prior work on bike sharing or bipartite matching (Angelopoulos et al. 2018; **?**; Vaze 2017), making them no longer applicable. For instance with classic budget feasible mechanisms such as (Singer 2010), the payment to a winning user is determined by the bids of other users and/or the budget. But in our case, this payment is invalid if it is above the value of the task assigned to the winning user.

In this paper, we propose a truth<u>F</u>ul and budg<u>E</u>t feas<u>I</u>ble

---

incentive mechanism with predicted $\underline{T}$ask r$\underline{E}$venue (FEIT-E) as the solution to the bike rebalancing problem, which we model as a reverse auction. In this auction, users bid for repositioning tasks and the platform determines task allocation and user payments. While we show that commonly used auctions and pricing mechanisms do not apply to our case, we specifically show that FEITE solves the problem and satisfies a number of desired economic and computational properties. We also prove two important theoretical guarantees backed by our mechanism. The performance of FEITE is then evaluated via simulations using a real industrial dataset obtained from a large bike-sharing company.

Our main contributions are summarized as follows:

- We model the bike rebalancing problem under a crowd-sourcing framework using an reverse auction model. Importantly, we incorporate a bipartite graph into the auction - with a payment constraint - to determine the allocation rule and the payment rule for the auction mechanism. This also implies that a bridge is set up between bike demand prediction and incentive mechanism design.

- We propose a mechanism FEITE which satisfies desired properties including incentive compatibility, budget feasibility, individual rationality, and computational efficiency. Notably, as is novel, we achieve this by combining Myerson's Lemma and a greedy weighted maximum matching technique.

- We prove two important theoretical guarantees: When the budget is tight, our mechanism ensures that the platform revenue is no less than its budget, under a practical large-market assumption. When the budget is sufficient, FEITE achieves a 2-approximation ratio as compared to the optimal solution which maximizes revenue; in addition, we show that the lower bound of the approximation ratio is $\sqrt{2}$ which is rather close to our ratio. Putting in practice perspectives, we provide a guideline of how to use this result to set company budget in real dynamic systems.

- We evaluate the effectiveness of our mechanism via extensive experiments using a real dataset from a large bike-sharing company. The results show that our mechanism outperforms other benchmark mechanisms in terms of revenue and profit.

Model

Consider a dynamic bike sharing system in which there is a batch of n users $N = \{1, 2, \ldots, n\}$ who have hired their bicycles and have not parked them (at the end of the hire) yet. There is also a set of m discrete locations[1] $L = \{1, 2, \ldots, m\}$ as bicycle parking lots, where each location can accommodate multiple bicycles but is conceptually considered a point on a map. We assume that the destinations of each user is known to the platform when she hires a bicycle, since it can be either reported by users as is a practice adopted by some companies (e.g., Hellobike in China),

_____

[1]We do not consider continuous locations because, nowadays, many countries such as Singapore and China have stipulated municipal regulations that shared bikes must be parked at designated locations rather than arbitrarily.

or predicted using historic information (Liu et al. 2018). The platform aims to incentivize the users to park their bicycles at system-desired locations for rebalancing the supply and demand of bicycles. Each user has a maximum relocation range h, out of which they would not accept a repositioning task; in other words, h is the maximum "extra mile" they are willing to relocate.

Similar to the concept of *first come first served flow* in (Waserhole, Jost, and Brauner 2013), we assume that a bicycle that is parked earlier will have a higher probability to be hired than a bicycle parked later. As such, each bicycle has a different probability of being hired even at the same location. Accordingly, we define a repositioning task t_{lx} as "park at location l as the x-th bicycle", which is associated with an expected revenue r_{lx}. As mentioned above, r_{lx} can be derived from the prediction of bike demand at this location (Yang et al. 2016; Li and Zheng 2019).

To address the task assignment problem, we construct a bipartite graph $G = \{N, T, E\}$, where the left nodes are the set of users N, and the right nodes are the set of tasks T. Since there are maximum n possible tasks at each location, there are totally $m \times n$ tasks in T. The set E is the edges connecting users and tasks and there is an edge between i and t_{lx} if location l is within the maximum relocation range of user i. For notation simplicity, henceforth we use j to denote a task t_{lx} when there is no ambiguity.

Each user i has a relocation cost c_i which is a private value only known to user i. She bids for a task with a claimed cost b_i which is not necessarily equal to c_i. Each task j has an expected revenue (or "value" as we use interchangeably) r_j for the platform. We aim to design an incentive mechanism that consists of an allocation rule and a payment rule, where the allocation rule specifies which task is allocated to which user, i.e., the matched pairs (i, j), and the payment rule specifies a payment p_i for each matched user i. The utility of a user is defined as $p_i - c_i$. In addition, the platform has a budget B for the rebalancing process, and the overall payment should not exceed the budget. As are desired, we want to design a mechanism that satisfies the following properties:

- *Incentive Compatibility*: a user can only maximize her utility by bidding truthfully, i.e., $b_i = c_i$. This property is also known as *truthfulness* or *strategy-proofness*.

- *Individual Rationality* for both users and platform: the payment to each winning (i.e., matched) user should be no less than her cost, i.e., $p_i \geq c_i$; the payment for each matched task should also be no more than its value, i.e., $p_i \leq r_j$, which is similar to the term *reserve price* in the theory of mechanism design.

- *Budget Feasibility*: the overall payment should be no more than the budget, i.e., $\sum_{(i,j) \in M} p_i \leq B$.

- *Computational Efficiency*: the mechanism should terminate in polynomial time.

Our objective is to maximize the platform revenue $R = \sum_{(i,j) \in M} r_j$, where $M = \{(i, j)\}$ is denoted as the set of matched user-task pairs. Thus, the problem of revenue max-

imization from task allocation can be formulated as

$$\mathbf{max} \quad R = \sum_{(i,j)\in M} r_j \quad\quad\quad (1)$$

$$\mathbf{s.t.} \quad p_i \leq r_j, \quad \forall (i,j) \in M$$

$$p_i \geq b_i, \quad \forall (i,j) \in M$$

$$\sum_{(i,j)\in M} p_i \leq B$$

where the first two constraints indicate the individual rationality for platform and users, and the third one indicates the budget feasibility.

We also evaluate platform profit which is defined as

$$Pr = \sum_{(i,j)\in M} r_j - \sum_{(i,j)\in M} p_i$$

and will be compared with revenue in our experiments.

Infeasibility of Existing Mechanisms

In this section, we show that some widely used mechanisms are not feasible for the bike rebalancing problem.

VCG mechnism. This is a classical mechnism that is strategy-proof and maximizes social welfare. However, it does not guarantee budget feasibility as required in our case. Proof by counter-example: see Figure 1 (a), where the platform has budget 1 and there are two users $\{a,b\}$ both with a small cost ϵ; the two tasks $\{1,2\}$ both have value 1 and there are two edges $\{(a,1),(b,2)\}$. The VCG mechanism will output matching $M = \{(a,1),(b,2)\}$ and the payment for each user is 1. Thus, the overall payment exceeds the budget and hence the mechanism is not budget feasible.
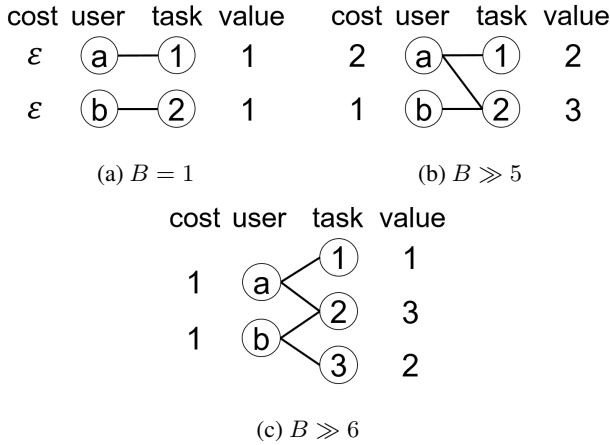

(a) $B = 1$ (b) $B \gg 5$

(c) $B \gg 6$

Figure 1: Counter examples for existing mechanisms.

Singer's mechanism. Proposed in (Singer 2010), this is another well-known mechanism yet is budget feasible. It greedily allocates user-task pairs with the highest ratio of r_j/b_i, but does not consider the individual rationality of the platform (i.e., $p_i \leq r_j$). If we adapt the mechanism by adding the constraint into the payment rule, it will no

longer guarantee incentive compatibility. To illustrate this, see Figure 1 (b), the platform has sufficient budget, and there are two users $\{a,b\}$ with costs $c_a = 2, c_b = 1$, two tasks $\{1,2\}$ with values $r_1 = 2, r_2 = 3$, and three edges $\{(a,1),(a,2),(b,2)\}$. The adapted mechanism outputs match $M = \{(a,1),(b,2)\}$, and $p_a = 2$. However, if user a misreports her cost to be a small number ϵ, she will be allocated task 2 and her payment becomes 3, making her better off.

Optimal matching. The third method is to choose a subgraph with $\sum r_j \leq B$ and then use the optimal (i.e., maximum) matching (that maximizes platform revenue) to allocate tasks, where the payments to winners are set to the values of their matched tasks. Not only does this mechanism has zero profit for the platform, but more importantly, it is also not truthful. To see this, consider Figure 1 (c), where the budget is sufficient, and there are two users $\{a,b\}$ with costs $c_a = 1, c_b = 1$, three tasks $\{1,2,3\}$ with values $r_1 = 1, r_2 = 3, r_3 = 2$, and four edges $\{(a,1),(a,2),(b,2),(b,3)\}$. The optimal matching will output match $M = \{(a,2),(b,3)\}$ since it achieves maximized revenue and the payments are $p_a = 3$ and $p_b = 2$. However, if user b untruthfully bids a cost of 3 instead of 1, then she will be assigned task 2 instead of 3 because task value must be no less than the cost. Therefore, she will receive a higher payment than bidding truthfully. Therefore, incentive compatibility is violated.

In fact, we will prove in Theorem 4 that there is no optimal mechanism that can satisfy the four properties simultaneously. Therefore, inspired by (Zhang, Wu, and Bei 2018), we propose an approximate mechanism (i.e., FEITE) in this paper.

Mechanism Design of FEITE

In this section, we present our proposed mechanism FEITE for the bike rebalancing problem. We first introduce a notion called *right-perfect matching* in a bipartite graph.

Definition 1 *A right-perfect matching in a bipartite graph G is a matching with size $|T|$, where T is the set of right nodes in graph G.*

In other words, we say a bipartite graph has a right-perfect matching if all tasks in the graph on the right can be matched to a user on the left. It is easy to see that a right-perfect matching is also a maximum matching of a bipartite graph.

The key idea of FEITE is to maintain a subgraph G' that always has a right-perfect matching. FEITE sorts all the tasks and users together in decreasing order of their values (costs), and then iterates over this sorted list. The mechanism tries to include more tasks with high values in G' and delete more users with high costs from it until the budget is exhausted or all elements are processed.

The complete pseudo-code of the mechanism FEITE is presented in Algorithm 1. In the mechanism, we maintain a variable of the remaining budget B' which is initially set as B. Once a user i is matched with payment p_i, we update B' as $B' - p_i$. In addition, we update a decreasing global price P during the algorithm process. Intuitively, for each

Algorithm 1: FEITE: a truthful and budget feasible incentive mechanism with predicted task revenue

---

Input: Bipartite graph $G = (N, T, E)$, budget B, cost c_i, value r_j, $\forall i \in N, \forall j \in T$.

Output: Task allocation $M = \{(i, j)\}$ and payment p_i for each winning user i.

1 Let $F = N \cup T$. For an element $e \in F$, if e is a user i, the value v_e is defined as her bid b_i; if e is a task j, the value v_e is defined as the expected revenue r_j.

2 Delete all edges (i, j) with $b_i > r_j$ in G.

3 $M \leftarrow \emptyset, B' \leftarrow B, T' \leftarrow \emptyset, N' \leftarrow \emptyset, E' \leftarrow \emptyset,$ $G' = (N', T', E')$;

4 Sort elements in F in decreasing order of v_e, breaking ties randomly, but if the tie is between a task and a user, let the task go first.

5 **for** *each element e in the above order* **do**

6   **if** *e is a task j* **then**

7   Let E_j be the set of incident edges of j in G that connect to unmatched users.

8   **if** *$G' \cup E_j$ has a right-perfect matching and*

9       $(|T'| + 1) \cdot r_j \leq B'$ **then**

10          $G' \leftarrow G' \cup E_j$.

11          $P \leftarrow r_j$.

12   **else**

13   Skip to next element.

14   **if** *e is a user i in G'* **then**

15   **if** *$G' \backslash i$ has a right-perfect matching* **then**

16          $G' \leftarrow G' \backslash i$.

17          $P \leftarrow b_i$.

18   **for** *Each user $i \in G'$* **do**

19   **if** *$G' \backslash i$ doesn't have a right-perfect matching* **then**

20   **for** *Each edge (i, j) of i in G'* **do**

21   **if** *$G' \backslash i \cup (i, j)$ has a right-perfect matching* **then**

22                $M \leftarrow M \cup (i, j), p_i \leftarrow P$.

23                $B' \leftarrow B' - p_i$.

24                $G' \leftarrow G' \backslash \{i, j\}$.

25   Skip to next user.

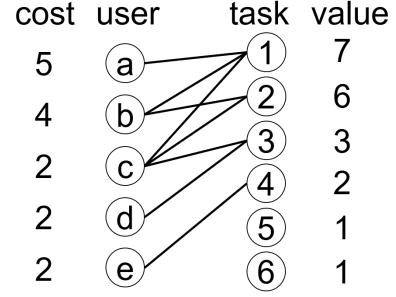26   **else**

27   Skip to next edge.

Figure 2: A walk-through example for our mechanism FEITE. Let the budget be $B = 14$. Tasks 1 and 2 are first considered, and they are added to G' with their affiliated edges since the budget is enough for them. So G' has three users a, b and c and none of them is critical for the right-perfect matching of G'. In addition, the global price $P \leftarrow 6$, and the remaining budget B' is still 14. Then, user a is removed from G' while updating the price $P \leftarrow 5$. Next, users b and c become critical for the right-perfect matching of G' now, and assume they are matched with task 1 and 2, respectively. The payment to each of b and c is $P = 5$, and hence we update $B' \leftarrow 4$. After that, task 3 will be added to G' with edge $(d, 3)$, we have $P \leftarrow 3$, and obviously d is critical and she will be allocated task 3 with payment 3. The remaining budget is also updated as $B' \leftarrow 1$. Finally, task 4 is considered but not added to G' because of the budget constraint, and tasks 5 and 6 are skipped because they have no edges. Therefore, the output matching is $\{(b, 1), (c, 2), (d, 3)\}$, and the revenue is 16 while the profit is 3.

Once the subgraph G' is changed (either a task is added or a user is removed), we check if there are critical users for the right-perfect matching of G', if so, for a critical user i, we allocate task j in G' to i if $G' \backslash i \cup (i, j)$ has a right-perfect matching. The payment is set as the global price P at this step, and then we update the remaining budget B'.

A walk-through example of FEITE is given in Figure 2.

Analysis of FEITE

Lemma 1 *The mechanism FEITE satisfies incentive compatibility.*

Proof *Since each user has only one private value (i.e., cost), this is a single-parameter problem and hence we can use the Myerson's lemma:*

Lemma 2 ((Myerson 1981)) *In single parameter auctions, for a normalized mechanism $\mathcal{M} = (f, p)$, where f is the allocation rule and p is the payment rule, $\mathcal{M}$ is incentive compatible iff it satisfies:*

1. ***Monotone allocation rule:*** *$\forall i \in N$, if $b'_i \leq b_i$, then $i \in f(b_i, b_{-i})$ implies $i \in f(b'_i, b_{-i})$ for every c_{-i};*

2. ***Threshold payment rule:*** *payment to each winning bidder is $\inf \{b_i : i \notin f(b_i, b_{-i})\}$.*

First, we prove the monotone allocation rule, i.e., once user i is matched by bidding b_i, she must be matched by bidding

element in the iteration of the sorted list, if it is a task j, the task and its affiliated edges (the connected user should be unmatched) will be added to G' if two conditions are satisfied after adding them: 1) G' still has a right-perfect matching and 2) the upper bound of the payment for all tasks in G' (i.e., $(|T'| + 1) \cdot r_j$) is below the remaining budget. If task j is added to G', we update the global price P as r_j. If the element is a user i, then if G' can maintain a right-perfect matching after discarding i (i.e., i is not *critical* for the right-perfect matching of G'), we then remove her from G', in this case, we also update the global price P as b_i.

$b'_i < b_i$. Let e be the element that updates the global price P as p_i. When bidding b_i, we use G_i^1 to denote the graph G' after e is processed (i.e., the step that P is updated as p_i), and G_i^2 has same definition while in the case of bidding b'_i. It's obvious that the algorithm process before P is updated as p_i is not affected by the bid of i. Therefore, we have $G_i^1 = G_i^2$, and user i is also critical for the right-perfect matching of G_i^2, hence the monotone allocation rule is satisfied.

Next, we prove the threshold payment rule, that is, if user i bids any cost larger than the payment p_i, she will not be matched, otherwise, she will be matched with a task. If $b'_i < p_i$, it can be observed that the value of all tasks in G_i^2 is higher than b'_i, thus the edges of i will not be deleted because $b'_i > r_j$. Therefore, similar to the proof of the monotone allocation rule, we have that $G_i^1 = G_i^2$, and user i is still critical, so she will be matched. If $b'_i > p_i$, user i will be considered before the element e. However, we can observe that, in the steps before element e is processed, user i is either not added to G' or is not critical for the right-perfect matching of G', otherwise she will be matched before element e. As a result, we have that user i will be discarded if $b'_i > p_i$, and the threshold payment rule is satisfied.

Lemma 3 *The mechanism FEITE is budget feasible.*

Proof We can observe that once a task j with value r_j is added to G', the payment to any user in G' is no more than r_j, since the global price P is non-increasing. Let j' be the last task added to G', $B'_{j'}$ the remaining budget before j' is added, and $|T_{j'}|$ the number of tasks in G' before adding j'. We have that

$$\sum_{(i,j)\in M} p_i \leq B - B'_{j'} + r_{j'} \cdot (|T_{j'}| + 1) \leq B$$

which concludes the proof.

We omit the proofs of the following two lemmas due to space constraint, please refer to our full paper [2].

Lemma 4 *The mechanism FEITE is individually rational for both users and platform.*

Lemma 5 *The mechanism FEITE satisfies computational efficiency.*

Theorem 1 *Our proposed mechanism FEITE is an incentive compatible, budget feasible, individually rational, and computational efficient mechanism.*

Theoretical Guarantee of Revenue

To show the theoretical guarantee of FEITE on revenue, we first introduce the large market assumption.

Assumption 1 (Large Market Assumption) *We   assume $c_i \ll B$ and $r_j \ll B$ for each user i and each task j.*

Intuitively, it's assumed that each individual user or task is negligible compared with the budget. This assumption is widely adopted in previous work (Vaze 2017; Anari, Goel, and Nikzad 2014) and it is practical in real world as the revenue of a single ride is indeed very small.

Next, we prove the theoretical guarantee under tight budget.

Theorem 2 *Under the large market assumption, we have $\sum_{(i,j)\in M} r_j \geq B$ if the budget is tight.*

Proof Let j_1 be the first task that is discarded because of budget constraint, $j0$ the last task added to G' before $j1$, and $|T_{j0}|$ ($|T_{j1}|$) the number of tasks in G' before considering $j0$ ($j1$). Assume that the set of tasks allocated between considering $j0$ and $j1$ is A_T, and the total payment for them is P_A. We have $r_{j0} \cdot (|T_{j0}| + 1) \leq B'_{j0}$ and $r_{j1} \cdot (|T_{j1}| + 1) > B'_{j1}$. Since the payments to users in A_T are all between r_{j0} and r_{j1}, we can get that

$$
\begin{aligned}
|T_{j1}| \cdot r_{j1} &\leq |T_{j1}| \cdot r_{j0} \\
&= (|T_{j0}| + 1) \cdot r_{j0} - |A_T| \cdot r_{j0} \\
&\leq (|T_{j0}| + 1) \cdot r_{j0} - P_A \\
&\leq B'_{j0} - P_A \\
&= B'_{j1}
\end{aligned}
$$

Combining the above inequations, we have

$$B'_{j1} - r_{j1} \cdot |T_{j1}| < r_{j1},$$

and further we obtain

$$
\begin{aligned}
\sum_{(i,j)\in M} r_j &\geq B - B'_{j1} + r_{j1} \cdot |T_{j1}| \\
&\geq B - r_{j1} \\
&\simeq B
\end{aligned}
$$

where the first inequation is because of the individual rationality of both platform and users and the last approximate equation is due to the large market assumption.

Before we prove the theoretical guarantee under sufficient budget, we first demonstrate that a greedy algorithm as shown in Algorithm 2 has an approximation ratio of 2, i.e., it can achieve at least half of the optimal revenue under sufficient budget.

Algorithm 2: A Greedy Mechanism

---
1 **for** *each task j in decreasing order of r_j* **do**
2   **for** *each edge (i,j) of task j* **do**
3   **if** *user i is not matched* **then**
4   Match i with j.
5   Skip to next task.

Lemma 6 *Algorithm 2 is a 2-approximation algorithm if the budget is sufficient.*

The proof is provided in our full paper due to space limitation.

Next, we prove the following lemma by showing that the allocation of our mechanism coincides with a particular run of the greedy algorithm.

Theorem 3 *The mechanism FEITE is a 2-approximation mechanism if the budget is sufficient.*

Proof *Since the budget is sufficient, we know that no task is discarded due to the budget limitation. We denote the matching in our mechanism as M. It's assumed that this matching is produced as following: For each task j in decreasing order of r_j, if $j \in M$, we allocate task j to its matched user in M, otherwise the task is skipped. Now we prove that, in the greedy algorithm, this process can also happen.*

It's obvious that for task $j \in M$, we can assign task j to its matched user in M in the greedy algorithm, thus we only need to prove that for each task $j \notin M$, when we consider it in the greedy algorithm, there is no unassigned user that has an edge to j.

Next, for contradiction, assume that we can find such an unmatched user i that has an edge to $j \notin M$ when processing j in the greedy algorithm. Then in our mechanism, when considering j, there will be a right-perfect matching for graph $G' \cup \{j\}$, i.e., $M \cup (i,j)$. Thus, the task will be added to G'. Note that in our mechanism, any task added to G' will end up being matched. This contradicts with our assumption and hence the lemma is proved.

To understand how "good" the approximation ratio of 2 is, next we prove that the lower bound is at least $\sqrt{2}$.

Theorem 4 *There is no mechanism that satisfies incentive compatibility and individual rationality can achieve better than $\sqrt{2}$-approximation when the budget is sufficient.*

Proof *We prove the lemma with a concrete counter example. Assume there exists a mechanism $\mathscr{F}$ that can achieve an approximation ratio better than $\sqrt{2}$. Consider case 1 where there are two users $\{a, b\}$, three tasks $\{1, 2, 3\}$ and 4 edges $\{(a,1),(a,2),(b,2),(b,3)\}$. In addition, we have that $c_a = \epsilon, c_b = \sqrt{2}+1, r_1 = 1+\epsilon, r_2 = \sqrt{2}+1+\epsilon, r_3 = \sqrt{2}+1$, where ϵ is a small positive number. We can observe that the optimal matching should be $\{(a,2),(b,3)\}$ which achieves revenue of $2\sqrt{2}+2+\epsilon$, we now prove that any mechanism that satisfies the above properties can achieve total revenue of at most $2+\sqrt{2}+2\epsilon$.*

First we consider case 2 where the only difference with case 1 is that $c_b = \sqrt{2}+1+\frac{\epsilon}{2}$ and hence the edge $(b,3)$ has to be deleted. In case 2, mechanism $\mathscr{F}$ can only output the matching $\{(a,1),(b,2)\}$, otherwise, the approximation can be at least $\frac{r_1+r_2}{r_2} > \sqrt{2}$. Moreover, due to individual rationality, $p_b \geq \sqrt{2}+1+\frac{\epsilon}{2}$.

Then we consider case 1, in the output matching of $\mathscr{F}$, if user b is matched with task 3, the payment is at most $1+\sqrt{2}$ and the utility of user b is at most 0. If user b misreports cost of $\sqrt{2}+1+\frac{\epsilon}{2}$, it becomes case 2, and as stated above, the utility of user b can be at least $\frac{\epsilon}{2}$. Thus, for incentive compatibility, mechanism $\mathscr{F}$ has to allocate task 2 to user b and pay at least $\sqrt{2}+1+\frac{\epsilon}{2}$. Hence the output matching of $\mathscr{F}$ can achieve revenue of at most $2+\sqrt{2}+2\epsilon$, and the approximation ratio limit is $\sqrt{2}$ when $\epsilon \to 0$.

Practical Implication: We now show how those results can be used when setting budget in a real dynamic system. The platform can set sufficient budget in the first time and obtain a revenue R_{suf}, then based on Theorem 3, we know

the optimal revenue is at most $2 \cdot R_{suf}$. After that, the platform is able to set a tight budget $B = \beta \cdot R_{suf}$, and Theorem 2 guarantees that it achieves at least $\beta/2$ of the revenue of optimal solution. This way, the platform can control the budget while there are not a lot of revenue loss.

Evaluation

We conduct simulation using a real-world dataset obtained from a large bike-sharing company in China called Mobike. The bike riding data cover 8×8 regions of Beijing with each region being 0.6km $\times$ 0.6km, and are dated from May 10th to 14th, 2017. With the same distribution of destinations in this dataset, we build a simulator which can randomly generate users' destinations. In the experiments, we set the number of users $n = 200$, and test different location numbers m. The cost of each user c_i is drawn from uniform distribution over $[0, \bar{c}]$ where $\bar{c} = 5$. The value of a task is calculated as the difference between the Kullback-Leibler (KL) divergences (Kullback and Leibler 1951) before and after fulfilling the task, similarly to previous work (Pan et al. 2019; Lv et al. 2019). Because of the space limitation, we refer the authors to (Lv et al. 2019) for concrete calculation of the task value. The acceptable range h is set as 300m and 600m, respectively. We also test the budget of 50 and 500 where 500 is sufficient while 50 is not. In addition, we conduct each experiment 10 times and take the average.

We compare FEITE with the following mechanisms:

- **APP-OPT:** As the budgeted matching problem is an NP-hard problem (it can be reduced to the knapsack problem), we cannot give an optimal allocation as benchmark. However, if the budget is tight and the large market assumption holds simultaneously, the following strategy is approximately an optimal mechanism: consider edges in decreasing order of r_j/b_i, match user i and task j if they are not ever matched before, and pay user i exactly her bid. The process stops until the budget is exhausted or there are no more edges. This mechanism achieves the maximum revenue under the above two assumptions but it's not truthful.

- **Greedy:** The greedy mechanism considers users in increasing order of their bids and allocates the task with the highest value as long as it is higher than the bid of the user. Once a user is matched, the price for all winning users is updated as the bid of her next user (whose bid is higher). The mechanism stops once a user cannot find a feasible task or the overall payment is above the budget, and pays all the winning users the bid of this unmatched user. If all users can be matched, we don't match the last one, and set her bid as the price for all other users. This greedy mechanism is incentive compatible but cannot provide any theoretical guarantee.

- **Surge:** Surge pricing is widely used in practice and it's an effective way to promote the revenue of platform (Guda and Subramanian 2019). We adopt a simple version of surge pricing here. Consider users in increasing order of their bids and allocate the task with the highest value to them once αr_j is higher than the bid of the user, where

(a) Revenue for $B = 50$ (b) Profit for $B = 50$

(c) Revenue for $B = 500$ (d) Profit for $B = 500$

Figure 3: Revenue and profit comparison when $h = 300m$.

(a) Revenue for $B = 50$ (b) Profit for $B = 50$
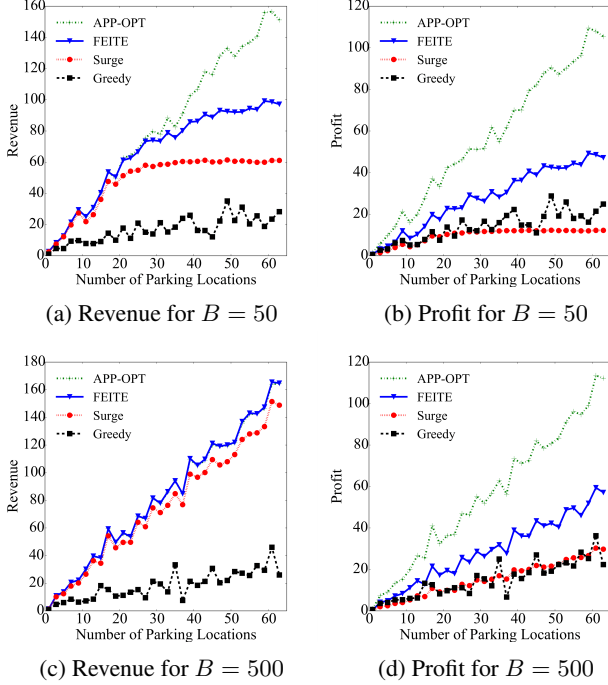
(c) Revenue for $B = 500$ (d) Profit for $B = 500$

Figure 4: Revenue and profit comparison when $h = 600m$.

$\alpha \in (0, 1)$ is a surge factor and r_j is the value of the task. The user is paid αr_j. The algorithm also terminates until the budget is exhausted or there are no feasible users. In the experiments, we set $\alpha = 0.8$. It can be easily shown that the Surge mechanism is not truthful.

Results

The experiment results for different acceptable extra distance $h = 300m$ and $h = 600m$ are given in Figure 3 and Figure 4, respectively. In Figure 3, we show the revenue and profit of platform under different parking location number m and different budget B. In general, as m increases, revenue and profit both increase, because of more right nodes in the bipartite graph. When the budget is tight ($B = 50$), we make the following specific observations. The revenue and profit of our mechanism are constantly higher than Surge and Greedy. When m is less than 30, as the bipartite graph is small, the budget is enough and our mechanism achieves nearly the same revenue as APP-OPT. When m is larger than 30, the budget will be exhausted and the revenue (and hence profit) of Surge mechanism stops increasing because it always has a constant ratio between payment and revenue. However, FEITE can output a better matching when the bipartite graph is larger and it shows much better performance over Surge and Greedy on both revenue and profit. When the budget is sufficient, our mechanism is not as good as APP-OPT but outperforms the others. As APP-OPT pays users exactly their costs, the mechanism is budget-saving but not practical in application because users will bid higher costs.

In Figure 4, h is larger so more edges appear in the bipartite graph. In this case, the performance of APP-OPT, FEIT-
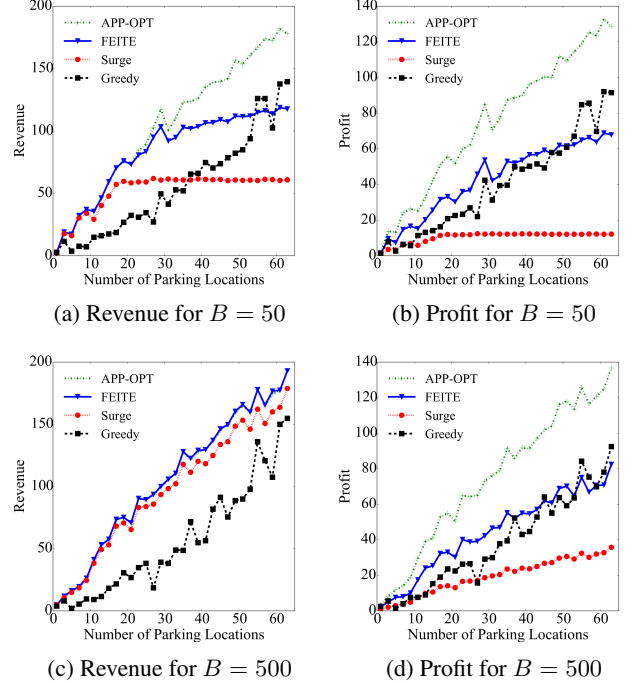
E and Surge are similar to Figure 3, but Greedy shows both much better revenue and profit. This is due to the fragility of Greedy mechanism as a single user can determine the termination of the mechanism. Thus, when $B = 300m$, the mechanism stops quickly because of the lack of edges, and when $B = 600m$, the termination is delayed. To show this, we test another 100 rounds of $m = 60$ for both Greedy and FEITE, and the maximum revenue of Greedy is 161.3, the minimum revenue is 19.7, and the variance is 648.1, while the data for FEITE is 127.2, 98.8 and 32.2 respectively. The profit shows similar results. Therefore, although Greedy can sometimes be slightly better than our mechanism, it is fragile and has much larger fluctuation.

In summary, our mechanism FEITE performs well on both revenue and profit, on top of its several desirable theoretical properties.

Conclusion and Future Work

In this paper, we have proposed an incentive mechanism to solve the bike rebalancing problem with predicted task value for bike sharing systems. It has desired theoretical properties as it satisfies incentive compatibility, budget feasibility, individual rationality and computational efficiency. In particular, we gave two theoretical guarantees under different budget constraints, and we showed how it can be applied in real dynamic systems. Its practical performance was evaluated using simulations based on real-world data, and the results demonstrate its superiority in terms of both revenue and profit. In our future work, we will explore tighter lower bounds and upper bounds for the problem. In addition, we plan to extend our algorithm into a real-time decision-

making mechanism while maintaining the theoretical properties, and conduct pilot studies in actual cities.

Acknowledgments

References

Anari, N.; Goel, G.; and Nikzad, A. 2014. Mechanism design for crowdsourcing: An optimal 1-1/e competitive budget-feasible mechanism for large markets. In *Proceedings of the 55th IEEE Annual Symposium on Foundations of Computer Science*, 266–275. IEEE.

Angelopoulos, A.; Gavalas, D.; Konstantopoulos, C.; Kypriadis, D.; and Pantziou, G. 2018. Incentivized vehicle relocation in vehicle sharing systems. *Transportation Research Part C: Emerging Technologies* 97:175–193.

Angeloudis, P.; Hu, J.; and Bell, M. G. 2014. A strategic repositioning algorithm for bicycle-sharing schemes. *Transportmetrica A: Transport Science* 10(8):759–774.

Fricker, C., and Gast, N. 2016. Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity. *Euro journal on transportation and logistics* 5(3):261–291.

Ghosh, S., and Varakantham, P. 2017. Incentivizing the use of bike trailers for dynamic repositioning in bike sharing systems. In *Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling*. AAAI Press.

Guda, H., and Subramanian, U. 2019. Your uber is arriving: Managing on-demand workers through surge pricing, forecast communication, and worker incentives. *Management Science* 65(5):1995–2014.

Kek, A. G.; Cheu, R. L.; Meng, Q.; and Fung, C. H. 2009. A decision support system for vehicle relocation operations in carsharing systems. *Transportation Research Part E: Logistics and Transportation Review* 45(1):149–158.

Kullback, S., and Leibler, R. A. 1951. On information and sufficiency. *The annals of mathematical statistics* 22(1):79–86.

Laporte, G.; Meunier, F.; and Calvo, R. W. 2015. Shared mobility systems. *4or* 13(4):341–360.

Li, Y., and Zheng, Y. 2019. Citywide bike usage prediction in a bike-sharing system. *IEEE Transactions on Knowledge and Data Engineering*.

Liu, Y.; Jia, R.; Xie, X.; and Liu, Z. 2018. A two-stage destination prediction framework of shared bicycles based on geographical position recommendation. *IEEE Intelligent Transportation Systems Magazine* 11(1):42–47.

Luo, T.; Kanhere, S. S.; Das, S. K.; and Hwee-Pink, T. 2016. Incentive mechanism design for heterogeneous crowdsourcing using all-pay contests. *IEEE Transactions on Mobile Computing* 15(9):2234–46.

Lv, H.; Wu, F.; Luo, T.; Gao, X.; and Chen, G. 2019. Hardness of and approximate mechanism design for the bike rebalancing problem. *Theoretical Computer Science*.

Maggioni, F.; Cagnolari, M.; Bertazzi, L.; and Wallace, S. W. 2019. Stochastic optimization models for a bikesharing problem with transshipment. *European Journal of Operational Research* 276(1):272–283.

Myerson, R. B. 1981. Optimal auction design. *Mathematics of operations research* 6(1):58–73.

Pan, L.; Cai, Q.; Fang, Z.; Tang, P.; and Huang, L. 2019. A deep reinforcement learning framework for rebalancing dockless bike sharing systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 1393–1400. AAAI Press.

Schuijbroek, J.; Hampshire, R. C.; and Van Hoeve, W.-J. 2017. Inventory rebalancing and vehicle routing in bike sharing systems. *European Journal of Operational Research* 257(3):992–1004.

Singer, Y. 2010. Budget feasible mechanisms. In *Proceedings of the 51st IEEE Annual Symposium on Foundations of Computer Science*, 765–774. IEEE.

Singla, A.; Santoni, M.; Bartók, G.; Mukerji, P.; Meenen, M.; and Krause, A. 2015. Incentivizing users for balancing bike sharing systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press.

Vaze, R. 2017. Online knapsack problem and budgeted truthful bipartite matching. In *Proceedings of the IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, 1–9. IEEE.

Waserhole, A.; Jost, V.; and Brauner, N. 2013. Vehicle sharing system optimization: Scenario-based approach. 2013b. *URL http://hal.archives-ouvertes.fr/hal-00727040*.

Yang, Z.; Hu, J.; Shu, Y.; Cheng, P.; Chen, J.; and Moscibroda, T. 2016. Mobility modeling and prediction in bikesharing systems. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, 165–178. ACM.

Zhang, J.; Zheng, Y.; Qi, D.; Li, R.; and Yi, X. 2016. Dnn-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 92. ACM.

Zhang, C.; Wu, F.; and Bei, X. 2018. An efficient auction with variable reserve prices for ridesourcing. In *Proceedings of the Pacific Rim International Conference on Artificial Intelligence*, 361–374. Springer.