An Auction Based Task Dispatching and Pricing Mechanism in Bike-sharing

Yaping Deng†, Bing Shi†‡*, Han Yuan†
†School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430000, China
Emails: {yapingdeng, bingshi, yuanh}@whut.edu.cn
‡Shenzhen Research Institute of Wuhan University of Technology, Shenzhen 518000, China

Abstract—As a green and low-carbon transportation way, bike-sharing provides lots of convenience in the daily life. However, how to dispatch bikes efficiently is a key issue in such a system. The bike-sharing platform can hire workers and pay to incentivize them to accomplish the dispatching tasks. However, there exist multiple workers competing for the dispatching tasks, and they may strategically report their task accomplishing costs (private information known by themselves) in order to make more profits, which may result in inefficient task dispatching. In this paper, we first design a dispatching algorithm named GDY-MAX to allocate tasks to workers. Furthermore, we design a strategy proof mechanism under the budget constraint to allocate tasks and determine the payments to workers. We theoretically prove that our mechanism can satisfy the properties of incentive compatibility, individual rationality and budget balance. Furthermore, we run extensive experiments to evaluate our mechanism based on a Mobike dataset. The results show that our approaches can make better performance than benchmark approaches.

I. INTRODUCTION

Bike-sharing as a low-carbon travelling way, can solve the problem of “the last mile” in the public transportation system. There exist many bike-sharing platforms, such as Mobike, HelloBike and so on, where Mobike has put over 7 million bikes in various regions of China. However, the redistribution problem emerges with the flourishment of the bike-sharing market. For example, bikes can be parked anywhere when riders reach their destinations, and then other riders cannot find the available bikes. The platform needs to dispatch these bikes regularly to specific locations in order to satisfy riders’ demands (e.g. dispatching nearby bikes to the exit of subway station) in regions. However, the bikes are usually distributed anywhere within a specific geographical area, and thus it is inconvenient and costive to use trucks to convey them. Actually, in the real world, some riders might be willingly to accomplish these dispatching tasks in their spare time to make money, and thus the bike-sharing platform can pay to incentivize these riders (i.e. workers) to dispatch bikes to specific locations. In this paper, we intend to analyze how the bike-sharing platform hires workers to dispatch bikes efficiently.

Specifically, the bike-sharing platform needs to hire workers to dispatch bikes to maximize the total values of accomplished dispatching tasks. Similar to some existing works [1], [2], we consider that the platform adopts an auction based mechanism to dispatch bikes and pay to workers. In this mechanism, the platform divides the overall area into a number of small regions. In each region, there exist a number of bikes to be dispatched. Workers bid by reporting their accomplishing costs of dispatching all bikes in that region, i.e. workers bid for the region including a set of tasks, not any individual tasks. The platform then determines how to match workers with regions, and determines the payments to the workers in order to maximize the total values of all regions. Multiple workers compete for the bike dispatching regions to make profits, and they may untruthfully report their costs of accomplishing tasks in order to make more profits. This may result in inefficient task dispatching. Furthermore, the platform usually has a budget constraint for paying to workers. Therefore, how to design an efficient auction mechanism for dispatching tasks and paying to workers under the budget constraint to maximize the total accomplishing values of regions is a key issue.

There exist some works about bike dispatching, such as [3]–[8]. However, they do not consider users’ strategic behavior. Specifically, this paper advances the state of art in the following ways. We first design a task dispatching algorithm, named GDY-MAX, to maximize the total value of regions given the budget constraint. The experimental results show that this task dispatching algorithm can achieve good performance. However, GDY-MAX cannot meet strategy proof. Then based on Myerson’s theorem [10], we further design a strategy proof mechanism where the task dispatching algorithm is monotonous and the worker pricing algorithm is based on critical price. We prove that this mechanism can satisfy the properties of incentive compatibility, individual rationality and budget constraint. We run extensive experiments to evaluate GDY-MAX and the strategy proof mechanism based on a Mobike dataset. The results show that the performance of the mechanism and GDY-MAX is similar to the optimal algorithm in the coverage ratio of accomplished task regions and the sum of task region values, and our mechanism has better performance than the uniform algorithm in the total payment.

The rest of the paper is structured as follows. We describe the basic settings in Section II. In Section III, we introduce GDY-MAX dispatching algorithm and the strategy proof

* Bing Shi is the corresponding author.

†Strategy proof means that workers have no incentives to misreport their private information, i.e. satisfying incentive compatibility and individual rationality [9].
mechanism. We run experiments to evaluate our methods in Section IV, and conclude the paper in Section V.

II. BASIC SETTINGS

In this section, we introduce the basic settings of tasks, workers, and provide the problem definition. As we have discussed previously, we divide the overall area into several small regions. Instead of bidding for the specific task, workers bid for the specific region. The platform determines the matching between the regions and workers, and computes the payment to workers. Since different regions may have different values, we intend to maximize the sum of region values given the budget constraint.

A. Task Region

In this paper, we divide the overall area $D$ into $k$ small hexagon regions $D = \{D_1, D_2, \ldots, D_k\}$. The task regions are represented by a set $T = \{t_1, t_2, \ldots, t_k\}$, where $t_j = (l_j, v_j)$, $l_j$ is the centric location of that region, and $v_j$ is the value of that region.

B. Workers

Workers bid for task regions, and then accomplish the dispatching tasks of the matched region. We use $M = \{m_1, m_2, \ldots, m_n\}$ to represent the set of workers, and use $\text{pos}_i$ to represent the location of worker $m_i$. The task region which can be accomplished by the worker depends on the geographical region $D_{m_i}$ and the worker’s activity radius $\text{rad}_i$. Now the set of task regions accomplished by worker $i$ is $T_i = \{t_j | \text{dis} (\text{pos}_i, l_j) \leq \text{rad}_i, t_j \in T\}$.

Each worker has a cost for accomplishing the dispatching tasks of that region, consisting of the time cost and energy cost, which is denoted by $c_i = \alpha \times H + \beta \times S$, where $H$ represents the total time to complete the task, $S$ represents the total walking distance, $\alpha$ and $\beta$ are time and distance parameters respectively. For worker $m_i \in M$, its expected utility $u_i$ is defined as the payment $p_i$ received from the platform minus the cost $c_i$, i.e. $\mu_i = p_i - c_i$.

When the platform publishes the task information, workers compete with each other to bid for the task region. The bid submitted by worker $m_i$ is denoted as $b_i = (T_i, c_i)$, including the set of regions $T_i$ that can be done by the worker and the total cost $c_i$. The bids of all workers are denoted as $b = (b_1, b_2, \ldots, b_M)$. Note that the worker may lie about cost $c_i$ (i.e. its bid may be different from the true cost) in order to make more profits.

C. Problem Definition

We use $M$ to represent the set of workers who win the regions, and then the assigned regions are denoted as the set $T_M = \cup_{m_i \in M} T_i$. Note that the payment should be greater than the worker’s cost, and the total payment should not be greater than the budget constraint $B$. We use $p = \{p_1, p_2, \ldots, p_M\}$ to represent the payments to all workers. We intend to maximize the overall region values, which is defined as follows.

**Definition 1 (Bike Task Value Maximization Problem)**

Given the set of task regions $T$, the set of workers $M$ and the set of bids $b$, the platform will dispatch task regions to workers $M \subseteq M$ (i.e. task dispatching algorithm) and make the payment to workers (i.e. pricing algorithm) under the budget constraint $B$, in order to maximize the sum of values $V(M) = \sum_{t_j \in T_M} v_j$, i.e.:

$$\max : V(M), \text{ s.t. } \forall m_i \in M, \sum_{m_i \in M} p_i \leq B$$

This problem is a typical set cover problem [11], which is an NP-hard Problem, and thus an approximation algorithm is needed. We adopt a submodular function in this paper.

**Definition 2 (Submodular functions)** $N$ is a finite set. Function $f : 2^N \rightarrow \mathbb{R}$ is a submodular function if and only if $S_1 \subseteq S_2 \subseteq N$ and $x \in N \setminus S_2$:

$$f (S_1 \cup \{x\}) - f (S_1) \geq f (S_2 \cup \{x\}) - f (S_2) \quad (1)$$

According to the definition of submodular function, for any set of workers $M_1 \subseteq M_2 \subseteq M$ and worker $m_i \in M \setminus M_2$, $M_1 \subseteq M_2$, there is $V (M_1) \leq V (M_2)$. The value is:

$$V (M_1 \cup \{m_i\}) - V (M_1) = \sum_{t_j \in T_1 \cup \{m_i\} \setminus T_k} v_j \quad (2)$$

Because of the diminished marginal effect, we have:

$$\sum_{t_j \in T_1 \cup \{m_i\} \setminus T_k} v_j \geq \sum_{t_j \in T_1 \setminus \{m_i\} \setminus T_k} v_j \quad (3)$$

According to Equations 2 and 3, we have:

$$V (M_1 \cup \{m_i\}) - V (M_1) \geq V (M_2 \cup \{m_i\}) - V (M_2) \quad (4)$$

III. DISPATCHING AND PRICING

In this section, we first describe the task dispatching algorithm GDY-MAX, and then introduce the strategy proof mechanism which ensure workers bid truthfully.

A. A Task Dispatching Algorithm (GDY-MAX)

We first design a greedy based dispatching algorithm to solve the dispatching issue. The reason of doing this is that the strategy proof mechanism is based on this algorithm. In addition, this algorithm can also achieve good performance in the experimental analysis. We call it as GDY-MAX, which includes GDY operation and MAX operation.

In more details, GDY operation chooses workers with low budget, in order to maximize the sum of values $V(M) = \sum_{t_j \in T_M} v_j$, i.e.:

$$\max : V(M), \text{ s.t. } \forall m_i \in M, \sum_{m_i \in M} p_i \leq B$$

Furthermore, we use MAX operation to choose the region having the largest value, i.e. $m^* \leftarrow \arg \max_{m_i \in M} V (\{m_i\})$. We then select the largest set of task regions by comparing
workers are denoted as a set \( V(M_k) \) with \( V(m^*) \), where \( m^* \) is the set of workers who can get the largest sum value of task regions.

Although the MAX operation overcomes the deficiency on the approximation of GDY operation, GDY-MAX algorithm is not monotonous. For example, worker \( m_i \in M_k \) with cost \( c_i \) reports its cost as \( c' \) and \( c' < c_i \). According to the unit margin contribution, when the cost of worker \( m_i \) is lower than a certain value, the set of workers selected by GDY-MAX algorithm could be \( \{m^*\} \), or \( \{m_i\} \). Therefore, GDY-MAX algorithm does not satisfy monotonicity, and thus is not strategy proof.

B. A Strategy Proof Mechanism

In this section, we intend to design a strategy proof mechanism (named BMTVMP) including task dispatching and pricing under the budget constraint to maximize the value of task regions while ensuring workers bid truthfully. According to Myerson’s theorem, the task dispatching algorithm in the mechanism needs to satisfy monotonicity, and the worker pricing algorithm should be based on the critical price.

1) Task Dispatching:
We adopt linear rounding [12] to address this monotonic issue. First, we compute \( V(\{m^*\}) \), and then compute the values of integer programming solution, named \( V \). The set with the largest values will be selected. The task dispatching algorithm can guarantee monotonicity with only a slight sacrifice of performance. In more details, for the linear rounding process of BMTVMP in the integer programming, BMTVMP \( (B/2, M^-) \) means that variable \( x_i \) (which shows whether the worker is selected by the platform as the winning worker) changes from a discrete value 0,1 to continuous value \([0,1]\) with the budget constraint changed to \( B/2 \). To solve the linear programming problem, worker \( m^* \) with the largest region value is eliminated since the final linear programming solution needs to be compared with worker \( m^* \). Workers with task accomplishing cost greater than \( B/2 \) should be excluded. Those workers are denoted as a set \( M_B \). Then the rest of workers is expressed as \( M^- \triangleq M \setminus \{m^*\} \cup M_B \). Finally, the optimal solution of linear programming can be denoted as GDY-LP-MAX \( (B/2, M^-) \). Similarly, the optimal solution of the integer programming is expressed as GDY-IP-MAX \( (B/2, M^-) \).

Similar to GDY-MAX algorithm, BMTVMP \( (B/2, M^-) \) is also based on greedy method. However, BMTVMP \( (B/2, M^-) \) has different MAX operation. The task dispatching algorithm compares the maximum region value with the value of the optimal linear programming solution GDY-LP-MAX \( (B/2, M^-) \). Instead of comparing values directly, we set a parameter \( \theta = \frac{6e^2}{(e-1)^2} \), which is related to the approximation ratio. The reason of setting such a parameter value is as follows. From the degree of similarity between GDY-LP-MAX \( (B/2, M^-) \) and \( V(M_k) \), we have:

\[
\text{GDY-IP-MAX} \left( \frac{B}{2}, M^- \right) \leq \frac{3e}{e-1} \max \{ V(M_k), V(m^*) \} \tag{5}
\]

According to pipage rounding [13] in the linear programming, we have:

\[
\text{GDY-LP-MAX} \left( \frac{B}{2}, M^- \right) \leq \frac{2e}{(e-1)^2} \text{GDY-IP-MAX} \left( \frac{B}{2}, M^- \right) \tag{6}
\]

Based on Equations 5 and 6, we have:

\[
\text{GDY-LP-MAX} \left( \frac{B}{2}, M^- \right) \leq \frac{6e^2}{(e-1)^2} \max \{ V(M_k), V(m^*) \} \tag{7}
\]

Therefore, the parameter \( \theta \) is set to \( \frac{6e^2}{(e-1)^2} \).

Finally, the task dispatching algorithm of BMTVMP mechanism is shown in Algorithm 1. Lines 3 to 6 indicate how to sort workers based on the marginal contributions. Then lines 8 to 11 compare \( V(m^*) \) with the optimal solution of linear programming GDY-LP-MAX \( (B/2, M^-) \). When GDY-LP-MAX \( (B/2, M^-) \) is lower than \( \theta \times V(m^*) \), we choose \( \{m^*\} \) as the winning worker, otherwise, we select \( M_k \) as the winner. The time complexity of this algorithm is \( O(|M|^2) \).

Algorithm 1: Task Dispatching Algorithm

\begin{itemize}
  \item [1] Task Dispatching:
  \item [2] \( M_k \leftarrow \emptyset \);
  \item [3] \( m^* \leftarrow \arg \max_{m \in M} V(\{m\}) \);
  \item [4] while \( M \setminus M_k \neq \emptyset \) and \( c_i^* \leq \frac{\theta V(M \cup \{m_i^*\})}{2V(M \cup \{m_i\})} \) do
  \item [5] \( M_k \leftarrow M_k \cup \{m_i\} \);
  \item [6] end
  \item [7] \( M \leftarrow M \setminus \{m^*\} \cup M_B \);
  \item [8] if GDY-LP-MAX \( (B/2, M^-) \) > \( \frac{6e^2}{(e-1)^2} \times V(\{m^*\}) \) then
  \item [9] \( M^* \leftarrow M_k \);
  \item [10] else
  \item [11] \( M^* \leftarrow \{m^*\} \);
  \item [12] return \( M^* \);
\end{itemize}

Lemma 1: The task dispatching algorithm of BMTVMP mechanism satisfies monotonicity.

\section*{proof 1:}
From lines 8 and 11 of Algorithm 1, we can see that there are two possible outputs. We now discuss them in turn

(1) If the output is \( M^* = \{m^*\} \), it shows that GDY-LP-MAX \( \left( \frac{B}{2}, M^- \right) \leq \theta \times V(m^*) \). In this case, the cost of worker \( m^* \) to complete the task only needs to satisfy the budget constraint \( B \). Therefore, the bid of worker \( m^* \) has no influence on the final result, which means that this worker will always be selected as the winner.

(2) If the output is \( M^* = M_k \), we have GDY-LP-MAX \( \left( \frac{B}{2}, M^- \right) \geq \theta \times V(m^*) \). Assuming that there is no change in task cost for the remaining workers other than worker \( m_i \), worker \( m_i \in M_k \) will submit new bid where the task cost will be reduced from \( c_i \) to \( c_i' \). In this case, the value of GDY-LP-MAX \( \left( \frac{B}{2}, M^- \right) \) will be increased due to the reduced cost of worker \( m_i \), satisfying the statement in line 8. For the marginal contribution, since the task cost is reduced to \( c_i' \), the position of worker \( m_i \) will be advanced in the overall workers. We assume that the new position is \( j \) and \( j < i \). Because of the decreased marginal effects, we have \( f_{i|M_k-1} \leq f_{i|M_k-1} \). The sum of values is, however, increased with the increased workers, which can be expressed...
as \( V(M_{i-1} \cup \{m_i\}) \geq V(M_j \cup \{m_i\}) \). In combination with budget constraint, we have
\[ c' \leq c_i \leq B \times \frac{f_{ij}M_{i-1}}{V(M_{i-1} \cup \{m_i\})} \leq B \times \frac{f_{ij}M_{j-1}}{V(M_j \cup \{m_i\})}. \]

Based on the above analysis, if the original winning worker reduces the task cost, it still satisfies the budget constraint and the marginal contribution condition, and thus is still selected as the winning worker. In summary, the task dispatching algorithm in BMTVMP mechanism satisfies monotonicity.

2) Pricing:

According to Algorithm 1, there will be two possible outputs. When the final result is worker \( m^* \) with the maximum value, the platform can take the entire budget as its payment. Otherwise, the platform needs to calculate the payment to each winning worker separately and ensure that the total payment does not exceed the budget constraint \( B \).

Definition 3 (Critical Cost). \( p \) is the worker’s critical task cost. If the task accomplishing cost reported by the worker is greater than its critical cost \( p \), the worker will not be assigned to the task, otherwise it will be assigned.

When the payment to worker \( m_i \) is calculated, we can remove \( m_i \) from the set of sorted winning workers. Then, the bid price that worker \( m_i \) wins at each position is calculated separately. We then choose the largest price as the payment to worker \( m_i \). This method calculates the payment \( p_i \) as the critical cost of worker \( m_i \in M^* \). Note that the payment to unassigned workers is set to 0, while the payment to worker \( m^* \) with the maximum value is set to \( B \). In the case of multiple winning workers, we first remove worker \( m_i \), and get the remaining workers \( M^{-i} = M \setminus m_i \). According to the greedy method, we compute the winning bid price which can defeat worker from first to \( k^{-i} + 1 \) position, denoted as \( c^i_{(j)} \). \( c^i_{(j)} \) has two limitations. First, the marginal contribution of worker \( m_i \)’s unit cost is greater than \( m_j \)'s unit cost:
\[ \frac{f_{ij}^{-i}}{c^i_{(j)}} \geq \frac{f_{ij}^{-i}}{c_j} \Rightarrow c^i_{(j)} \leq \frac{f_{ij}^{-i} \times c_j}{f_{ij}^{-i}} = \varphi^i_{(j)} \] (8)
\[ c^i_{(j)} \leq B \times \frac{f_{ij}^{-i}}{V(M_{j-1} \cup \{m_i\})} = \lambda^i_{(j)} \] (9)

From Equations 8 and 9, we know that \( c^i_{(j)} \leq \min\{\varphi^i_{(j)}, \lambda^i_{(j)}\} \). The pricing algorithm use the smallest value between \( \varphi^i_{(j)} \) and \( \lambda^i_{(j)} \) as the payment \( p^i_{(j)} \) for worker \( m_i \). When the position index \( j \) increases, \( f_{ij}^{-i} \) is monotonically increasing, and \( \frac{f_{ij}^{-i}}{c^i_{(j)}} \) is monotonically decreasing according to the submodularity of the value function. Therefore, we can select the maximum value of \( p^i_{(j)} \) and set it as the payment to \( m_i \).
\[ p_i = \max_{1 \leq j \leq (k^{-i} + 1)} p^i_{(j)} \] (10)

The pricing algorithm is shown in Algorithm 2.

\begin{algorithm}
\caption{Worker Pricing Algorithm}
\begin{algorithmic}[1]
\State \textbf{Input:} Set of workers \( M \); Set of task regions \( T \); Budget constraint \( B \); Set of bidding \( b \); Set of winning workers \( M^* \)
\State \textbf{Output:} Payment \( p \)
\Function{Proof}{2}
\If{$M^* = \{m^*\}$}
\State $p \leftarrow 0$;
\EndIf
\If{$M^* = \emptyset$}
\State $p \leftarrow B$;
\EndIf
\For{$m_i \in M^*$}
\State $M^{-i} \leftarrow M \setminus \{m_i\}$, $j \leftarrow 1$, $M_j^{-i} \leftarrow \emptyset$, $p_i \leftarrow c_i$;
\State $m_j \leftarrow \arg\max_{m_i \in M^{-i}} \frac{f_{ij}^{-i}}{c_i}$;
\EndFor
\For{$M^{-i} \setminus M_j^{-i-1} \neq \emptyset$}
\State $c_j \leftarrow \max\{\min\{\varphi^i_{(j)}, \lambda^i_{(j)}\}, f_{ij}^{-i} \}$;
\State $p_i \leftarrow \max\{p_i, \min\{\varphi^i_{(j)}, \lambda^i_{(j)}\}\}$;
\State $M^{-i} \leftarrow M_j^{-i-1} \cup \{m_j\}$, $j \leftarrow j + 1$;
\State $m_j \leftarrow \arg\max_{m_i \in M^{-i} \setminus M_j^{-i-1}} \frac{f_{ij}^{-i}}{c_i}$;
\EndFor
\State $p \leftarrow \max\{p_i, \min\{\varphi^i_{(j)}, \lambda^i_{(j)}\}\}$;
\EndFunction
\end{algorithmic}
\end{algorithm}

Lemma 2: The payment calculated by the pricing algorithm meets the critical cost requirement.

\begin{proof}
Proof can be shown in two cases:
(1) Assuming that worker \( m_i \) submits a new bid, where the task cost of \( m_i \) is less than the critical cost \( p_i \), we have \( c_i \leq \theta_i(t) \). The worker can be assigned to the task at position \( r \).
(2) After worker \( m_i \) submits a new bid, where the task accomplishing cost is greater than or equal to the critical cost \( p_i \), the platform will no longer select the worker as the winner.

We discuss it in the following two cases.

When \( \theta_i(t) < \lambda_i(t) \), we get \( p_i(t^*) > \theta_i(t) \) from \( p_i \leftarrow \min\{\theta_i(t), \lambda_i(t)\} \). When \( m_i \)’s task cost is not less than \( p_i \), \( c_i > \theta_i(t) \) can be obtained. The cost of \( m_i \) increases while the marginal contribution remains, which means the marginal contribution per unit cost of worker \( m_i \) is smaller than the marginal contribution per unit cost of worker \( m_i \). At the same time, we sort the workers by marginal contribution of unit cost. Let \( j \in [r + 1, k^{-i} + 1] \) be below position \( r \), we have \( \theta_i(t) \geq \theta_i(t) \). Then we have \( c_i > \theta_i(t) > \theta_i(t) \), and \( m_i \) will not be selected as the winner. Suppose at position \( j \), we have \( \varphi^i_{(j)} > p^i_{(j)} > \varphi^i_{(j)} > \lambda^i_{(j)} \) from \( \theta_i(t) < \theta_i(t) \).
\end{proof}
the cost of worker $m_i$ has increased, $c_i > \theta_{i(j)} > \lambda_{i(j)}$, and it means that the budget constraint is not satisfied and $m_i$ is not selected as the winner.

When $\theta_{i(j)} > \lambda_{i(r)}$, $p_{i(r)}^{-i} = \lambda_{i(r)}$ could be deduced from $p_i \leftarrow \min \{\theta_{i(j)}, \lambda_{i(j)}\}$. Assuming at position $j \in [0, k^{-i} + 1]$, then $\lambda_{i(r)}$ is greater than $\lambda_{i(j)}$. $m_i$‘s cost increases $c_i > p_{i(r)}^{-i} = \lambda_{i(r)} > \lambda_{i(j)}$, and it will not be selected as the winner because of the budget constraint. Assuming that $\lambda_{i(r)} < \lambda_{i(j)}$, we can get $\lambda_{i(j)} > p_{i(r)}^{-i} = \lambda_{i(r)} > p_{i(j)}^{-i} = \varphi_{i(j)}$. Furthermore, we have $c_i > p_{i(r)}^{-i} = \lambda_{i(r)} > \varphi_{i(j)}$, it has conflict with $\theta_{i(r)} > \lambda_{i(r)}$, $m_i$ cannot be selected.

In conclusion, the payment $p_i$ calculated by the worker pricing algorithm is the critical cost of $m_i$, and BMTVMP mechanism satisfies incentive compatibility.

3) Theoretical Analysis:

In this section, we theoretically prove that our mechanism can satisfy individual rationality and budget constraint.

Theorem 1: BMTVMP mechanism satisfies incentive compatibility under budget constraint.

Proof 3: From Lemma 1 and Lemma 2, we know that the task dispatching algorithm is monotonous and the payment is the critical cost of the worker. According to Myerson’s theorem, BMTVMP mechanism satisfies incentive compatibility.

Theorem 2: BMTVMP mechanism satisfies individual rationality under budget constraint.

Proof 4: In worker pricing algorithm, the worker who does not participate in the dispatching task is paid zero. According to $p_i \leftarrow \max \{p_i, \min \{\theta_{i(j)}, \lambda_{i(j)}\}\}$, we know that the payment to $m_i$ is $p_i \geq p_{i(j)}^{-i}$. Bids submitted by other workers are unchanged when calculating the remuneration of $m_i$. In this case, other winning workers will remain unchanged except $m_i$, which can be represented by the set $M_{i-1} = M_{r-1}^i$. Worker $m_j$ could win by eliminating $m_i$. In the list that does not exclude $m_i$, worker $m_i$ is selected by the platform as the winner, which indicates that $m_i$’s task cost meets the budget constraint, and it can be inferred that:

$$c_i \leq \frac{B \times f_{i|M_{i-1}}^{-i}}{2V(M_{i-1} \cup \{i\})} = \frac{B \times f_{i|M_{i-1}}^{-i}}{2V(M_{r-1}^i \cup \{i\})} = \lambda_{i(j)} \quad (11)$$

The list is sorted by the marginal contribution of unit cost, and the position of worker $m_i$ precedes that of worker $m_j$: $f_{i|M_{i-1}}^{-i} / c_i \geq f_{j|M_{i-1}}^{-i} / c_j \Rightarrow c_i \leq \frac{f_{i|M_{i-1}}^{-i} \times c_j}{f_{j|M_{i-1}}^{-i}} = \varphi_{i(j)} \quad (12)$

From Equations 11 and 12, we have the task accomplishing cost of worker $m_i$:

$$c_i \leq \min \{\varphi_{i(j)}, \lambda_{i(j)}\} \leq p_{i(j)}^{-i} \leq p_i \quad (13)$$

The above equations indicates that the task accomplishing cost of worker $m_i$ is less than or equal to payment $p_i$. Then the utility of worker $m_i$ is greater than or equal to zero. To sum up, in either cases, the utility obtained by the worker is non-negative. Therefore, BMTVMP mechanism satisfies individual rationality.

Theorem 3: Given the dispatching task assigned to worker $m_i \in M^*$, the payment $p_i$ limited to $\frac{f_i}{V(M^*) \times B}$, the total payment calculated by BMTVMP mechanism meets the budget constraint.

Proof 5: Assuming that the output of the task dispatching algorithm is $M^* = \{m^*\}$. Now the worker’s pricing algorithm sets $B$ as the payment, which meets the budget constraint. In another case, the output of the task dispatching algorithm is $M^* = M_k$. Assuming that the worker pricing algorithm gives $m_i \in M_k$ a large payment, which satisfies $p_i > \frac{f_i}{V(M^*) \times B}$. Let position $j \in [0, k^{-i} + 1]$, the equation $r = \arg \max_{i(j)} p_{i(j)}^{-i}$ is established. As the payment $p_i$ is limited by the marginal contribution of unit cost and the budget, we have:

$$p_i \leq \frac{f_{i|M_{i-1}^i}^{-i}}{f_{i|M_{i-1}^i}} \cdot \frac{c_r}{2V(M_{r-1}^i \cup \{i\})} \quad (14)$$

According to individual rationality, the utility obtained by worker is non-negative, and $c_i \leq p_i$. Because worker $m_i$ is not selected within the former $(i - 1)$ positions, when $0 \leq j \leq (i - 1)$, we have $p_{i(j)}^{-i} < c_i$. Then we have $p_{i(j)}^{-i} < p_{i(j)}^{-i}$. Furthermore, $M_{i-1} \in M_{r-1}^i$, when $M_{r-1} \cup \{m_i\} = M_{r-1}^i \cup M_k$, it can be concluded that:

$$\frac{f_{i|M_{i-1}^i}^{-i}}{p_i} \geq \frac{f_{i|M_{i-1}^i}^{-i}}{p_i} \geq \frac{2V(M_{r-1}^i \cup \{m_i\})}{B} = \frac{2V(M_{r-1}^i \cup M_k)}{B} \geq \frac{V(M_k)}{B} \quad (15)$$

From Equation 15, it can be obtained that $p_i \leq \frac{f_i}{V(M^*) \times B}$. The original assumption is not true. Therefore, the payment $p_i$ given by the worker pricing algorithm to worker $m_i \in M_k$ meets $p_i \leq \frac{f_i}{V(M^*) \times B}$. We have that:

$$\sum_{m_i \in M^*} p_i \leq B \times \sum_{m_i \in M^*} \frac{f_i}{V(M^*)} = B \quad (16)$$

Therefore, the total payment of BMTVMP mechanism will not exceed the budget constraint.

IV. EXPERIMENTAL ANALYSIS

We now run experiments to evaluate our methods. In the experiment, we use a Mobike dataset including worker tracks in Shanghai from August 1 to September 1, 2016. Each data record contains order ID, bike ID, worker ID, start time, start location (longitude and latitude), end time, end location, and trace. The total amount of orders is 102361. In the experiment, we assume that the bikes needs to be dispatched periodically.

A. Experimental Settings

We first pre-process all order data and distinguish them into weekdays and weekends, as shown in Figure 1. We can see that there is a huge difference between the number of
orders on weekdays and weekends. In order to make a reliable and consistent experimental results, we exclude the order data of weekends, and run the experiments on the order data of weekdays.

The map is divided into a number of small hexagon regions with two kilometers radius. We categorize the regions into 5 levels, and each level corresponds to different unit cost information. The worker’s radius of activity is divided into 5 levels, namely 1.5, 2, 2, 3 and 3.5 kilometers. We set 2 twice because there exist plenty of regions with 50 to 100 bikes. The worker cost is calculated by the base price plus the total price. The total price is obtained by multiplying the unit price with the number of bikes. The unit price is randomly selected from [0.01, 0.1] and the basic price is set [10, 20] at random. Furthermore, in the experiment, we set the number of workers between 50 and 600 with step size 50. For each worker, the set of accomplishing task region depends on radius, and the cost of task accomplishment is uniformly distributed. The budget is set between 500 and 2000 with step size 250. The experimental parameters are shown in Table I.

Furthermore, we evaluate our mechanism and GDY-MAX dispatching algorithm against two benchmark algorithms.

**UNIFORM** [14]: It sets the probability of 2/5 to assign the single worker with the maximum value, and the probability of 3/5 to choose the workers according to greedy algorithm. This probabilistic selection method enables the UNIFORM algorithm to have a constant approximation ratio in the worst environment. The comparison between **UNIFORM** algorithm and **BMTVMP** mechanism can intuitively shows the influence of payments on the maximization of task region values.

**OPTIMAL**: The optimal algorithm of the set coverage problem can obtain the optimal solution for the value maximization problem. Therefore **OPTIMAL** algorithm is selected. However, it cannot be used in real life since the computation is heavy and cannot guarantee strategy proof.

We evaluate **BMTVMP** and **GDY-MAX** against the above two algorithms in terms of the following metrics.

- The coverage ratio of accomplished task regions, which is the proportion of the regions completed by workers to the total number of regions.
- The sum of task region values, which is the accumulation of values of accomplished task regions.
- The total payment, which is paid to workers by the platform.

### B. Experimental Results

The experiments are repeated for 100 times, and we compute the average results for the analysis. We now discuss the experimental results in detail.

**The coverage ratio of accomplished task regions**: The coverage ratio of the four algorithms are shown in Figure 2, where the number of workers is set to 300, and the corresponding budget constraint is set from 500 to 2000. From Figure 2, we can see that the coverage ratios of the accomplished task regions of **GDY-MAX**, **OPTIMAL** algorithm and **BMTVMP** mechanism increase with the increased budget since the platform is able to hire more workers to complete more tasks, and thus increasing the regional coverage.

In general, three algorithms can achieve more than 70% coverage given the high budget except **UNIFORM** algorithm.
Furthermore, the performance of GDY-MAX algorithm is very similar to that of OPTIMAL algorithm. In contrast, the task coverage of BMTVMP mechanism is slightly lower, but the ratio of coverage growth is stable.

Note that the UNIFORM algorithm shows that the coverage tends to decline when the budget increases. It may be caused by the random selection of winning workers with a certain probability. Workers with high cost are selected to spend more budget and the amount of accomplished task regions is reduced, thus resulting in the decreased coverage ratio. On the other hand, we find that when the budget increases to a high value, BMTVMP mechanism is still unable to cover all regions. This is because the workers’ locations are randomly selected and thus cannot cover all regions.

Figure 3 shows the coverage ratio of four algorithms when the budget is set to 1000 and the number of workers is set from 50 to 600. The experimental results show that the overall coverage of GDY-MAX algorithm, OPTIMAL algorithm and BMTVMP mechanism increase with the increased number of workers. Compared with Figure 2, the coverage ratio increases faster, implying that the number of workers has a greater impacts on the coverage ratio than the budget.

The sum of the task region values: Figure 4 shows the sum of task region values with budgets from 500 to 2000 when the number of workers is 300. We can see that when the budget increases, the sum of task region values increase for all algorithms. The reason is that when the budget increases, bike-sharing platform can hire more workers to complete more tasks. When the budget exceeds 1000, GDY-MAX and OPTIMAL algorithms can almost complete all tasks and obtain high values compared with other algorithms. Note that when the budget reaches a relatively large value, BMTVMP mechanism can not only prevents the strategic behavior, but also achieves near-optimal performance.

Figure 5 shows the sum of task region values when the budget is 1000 and the number of workers varies between 50 and 600. Task values obtained by those four algorithms go up with the increased workers. When the number of workers increases, bike-sharing platform can select workers with lower cost to complete tasks, but still win the same value. We can see that BMTVMP mechanism can achieve similar performance to the optimal algorithm. Although BMTVMP mechanism has a loss of small amount of values, it can prevent strategic behavior.

The total payment: Since GDY-MAX algorithm and OPTIMAL algorithm do no include the pricing algorithm, we only evaluate the total payment of BMTVMP mechanism against UNIFORM algorithm. The number of workers is fixed at 300 and the budget ranges from 500 to 2000. The results are shown in Figure 6. We can see that the total payment increases with the increased budget. When the number of workers is large and the budget is increased, the bike-sharing platform can use more budget to hire more workers to complete tasks. The total payment of BMTVMP mechanism is less than UNIFORM algorithm, which implies that BMTVMP mechanism
can save budget and pay to workers more reasonably.

Next, the budget is fixed at 1000 and the number of workers is set between 50 and 600. The total payments of the bike-sharing platform in BMTVMP mechanism and UNIFORM are shown in Figure 7. The results show that the total payments of BMTVMP mechanism and UNIFORM algorithm decrease with the increased workers. The reason is that when the budget is fixed, the competition among workers becomes fierce when the number of workers increases. The platform will choose workers with lower cost to complete tasks, reducing the total payment. Similarly, the results show that the total payment of BMTVMP mechanism is less than UNIFORM algorithm. Compared with Figure 6, the total payments of both mechanisms decrease, implying that the number of workers has a greater impact on the total payment than the budget constraint.

V. CONCLUSION

In this paper, we first design a task dispatching algorithm GDY-MAX to allocate tasks to workers to maximize the value of accomplished task regions. Since this algorithm cannot prevent strategic behavior, we further propose a strategy proof mechanism to maximize the values of task regions under the budget constraint. We design a task dispatching algorithm satisfying the monotonicity, and design a worker pricing algorithm to prevent strategic behavior, we further propose a strategy proof algorithm. Since this algorithm cannot choose workers with lower cost to complete tasks, reducing the total payment. Similarly, the results show that the total payment of BMTVMP mechanism is less than UNIFORM algorithm. Compared with Figure 6, the total payments of both mechanisms decrease, implying that the number of workers has a greater impact on the total payment than the budget constraint.

ACKNOWLEDGMENT

This paper was funded by the Humanity and Social Science Youth Research Foundation of Ministry of Education (Grant No. 19YJC790111), the Philosophy and Social Science Research Foundation of Ministry of Education (Grant No. 18JHQ060) and Shenzhen Fundamental Research Program (Grant No. JCYJ20190809175613332).

REFERENCES

[1] M. H. Cheung, R. Southwell, F. Hou, and J. Huang, “Distributed time-sensitive task selection in mobile crowdsensing,” in The 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2015, pp. 157–166.

[2] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, “Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing,” in IEEE Conference on Computer Communications, 2014, pp. 1231–1239.

[3] S. V. E, J. Park, and Y. Cho, “Using data mining techniques for bike sharing demand prediction in metropolitan city,” Comput. Commun., vol. 153, pp. 353–366, 2020. [Online]. Available: https://doi.org/10.1016/j.comcom.2020.02.007

[4] M. Holienčinová, Z. Kádeková, T. Holota, and L. Nagyová, “Smart solution of traffic congestion through bike sharing system in a small city,” MONET, vol. 25, no. 3, pp. 868–875, 2020. [Online]. Available: https://doi.org/10.1007/s11036-020-01516-4

[5] L. Kazemi, C. Shahabi, and L. Chen, “GeoTrucrowd: trustworthy query answering with spatial crowdsourcing,” in 21st SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2013, Orlando, FL, USA, November 5-8, 2013, C. A. Knoblock, M. Schneider, P. Kröger, J. Krumm, and P. Widmayer, Eds. ACM, 2013, pp. 304–313. [Online]. Available: https://doi.org/10.1145/2525314.2525346

[6] D. Deng, C. Shahabi, and U. Demiryurek, “Maximizing the number of worker’s self-selected tasks in spatial crowdsourcing,” in 21st SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2013, Orlando, FL, USA, November 5-8, 2013, C. A. Knoblock, M. Schneider, P. Kröger, J. Krumm, and P. Widmayer, Eds. ACM, 2013, pp. 314–323. [Online]. Available: https://doi.org/10.1145/2525314.2525370

[7] Y. Singer, “Budget feasible mechanisms,” in 51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA. IEEE Computer Society, 2010, pp. 765–774. [Online]. Available: https://doi.org/10.1109/FOCS.2010.78

[8] N. Anari, G. Goel, and A. Nikzad, “Mechanism design for crowdsourcing: An optimal 1-1/e competitive budget-feasible mechanism for large markets,” in 55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014. IEEE Computer Society, 2014, pp. 266–275. [Online]. Available: https://doi.org/10.1109/FOCS.2014.36

[9] M. A. Satterthwaite and H. Sonnenschein, “Strategy-proof allocation mechanisms at differentiable points,” The Review of Economic Studies, vol. 48, no. 4, pp. 587–597, 1981.

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

[11] G. Bergantiños, M. Gómez-Rúa, N. Llorca, M. Pulido, and J. Sánchez-Soriano, “Allocating costs in set covering problems,” European Journal of Operational Research, vol. 284, no. 3, pp. 1074–1087, 2020.

[12] Y. Azar and I. Gamzu, “Truthful unification framework for packing integer programs with choices,” inInternational Colloquium on Automata Languages, and Programming, 2008, pp. 833–844.

[13] A. A. Ageev and M. I. Sviridenko, “Pipage rounding: A new method of constructing algorithms with proven performance guarantee,” Journal of Combinatorial Optimization, vol. 8, no. 3, pp. 307–328, 2004.

[14] Z. Zheng, F. Wu, X. Gao, H. Zhu, S. Tang, and G. Chen, “A budget feasible incentive mechanism for weighted coverage maximization in mobile crowdsensing,” IEEE Transactions on Mobile Computing, vol. 16, no. 9, pp. 2392–2407, 2016.