Received March 31, 2021, accepted April 14, 2021, date of publication April 20, 2021, date of current version June 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3074150

Online Task Scheduling With Workers Variabilities in Crowdsourcing

QI LI AND LIJUN CAI
College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
Corresponding author: Qi Li (qili@hnu.edu.cn)

ABSTRACT Crowdsourcing system, which utilizes many workers to process computer-complexity tasks, has become an effective platform in today's online labor markets. In a crowdsourcing system, maximizing the total utility is one key design goal. This goal is extremely hard because a computer-complexity task can be a multi-dimensional large-scale task that contains thousands or millions of atomic tasks. In online situation, we need to consider both the varying service of workers and future unknown task arrivals. As we know, none of the previous work considers a multi-dimensional large-scale task assignment for utility maximization. In this paper, an online framework is proposed to solve this optimization problem by running atomic tasks in parallel on workers. To estimate worker service rates, we consider each varying worker as an arm for a multi-armed bandit in crowdsourcing system. We design the online scheduling algorithm from a bandit perspective by Online Convex Optimization (OCO) techniques. We prove that our designed algorithm can yield a sublinear regret bound. Finally, we show that our designed algorithm is better than the baseline algorithms by nearly 10% for the total utility achieved.

INDEX TERMS Multi-armed bandit, online optimization, multi-dimensional large-scale crowdsourcing tasks, regret bound.

I. INTRODUCTION

Crowdsourcing system has become more and more popular for workers to perform computer-complexity tasks. In Amazon Mechanical Turk [1], translating Chinese papers into English papers, collecting school addresses and labelling contents of images are typical computer-complexity tasks. This typical task is usually a large-scale task which contains thousands or millions of atomic tasks. These atomic tasks are viewed as decomposed tasks such as binary choice or simple voting [2]. Moreover, this typical task can also be a multi-dimensional crowdsourcing task (data with multiple attributes) which can be used to maximize the total utility in this platform [3]. In a crowdsourcing system, tasks are assigned to workers and workers get payment after completing tasks. Note that the task payment is considered as task utility and we will use the term utility in the rest of paper. We utilize a concave utility function to calculate task utility with respect to the completed task [4]. The concave utility function can be used to consider task dependencies. In this paper, we utilize all A-tasks (atomic tasks) of a L-task (multi-dimensional large-scale crowdsourcing task) to maximize total utility, which are assigned to multiple workers and run in parallel [5].

In this paper, maximizing total utility is one key design goal, which is a hard optimization problem in an online manner. The hardness of this optimization problem comes from four aspects: (1) It is challenging to assign all L-tasks to workers under worker's capacity constraints and task's deadline constraints [6] (2) A task assignment decision is an integer programming problem [7]. Without knowing future task arrivals, it is difficult to achieve this goal by choosing a proper worker to complete a task. (3) A worker has varying skill by the time [8], [9]. Actually, the service of a worker may change significantly during the processing of a L-task. Thus, this situation leads to a large variation for the process of A-tasks which are contained in the same L-task. (4) It is not tractable to analyze designed algorithm's performance, because every online assignment will affect the remaining online assignments [10].

To maximize the total utility, a practical task assignment plan should meet the following four requirements: online manner task assignments, multi-dimensional tasks, concave utility functions, and the varying service of workers. As we

The associate editor coordinating the review of this manuscript and approving it for publication was Rashid Mehmood.

know, no prior work satisfies the four requirements. In [12], the scheme only satisfies the varying service of workers. In [9], the scheme is in an online manner and satisfies the varying service of workers. In [3], the scheme satisfies online manner task assignments and multi-dimensional tasks. In [13], the scheme satisfies online manner task assignments and concave utility functions.

In our model, a scheduling plan for L-tasks is proposed to maximize its total utility in a crowdsourcing system. To achieve online task assignments, we propose an online framework to solve this optimization problem. To satisfy capacity constraints, we use the short-term constraint to handle the online framework in slotted time. To handle the varying service of workers, we consider each varying worker as an arm for a multi-armed bandit in the system. To solve this MAB problem, we utilize the past service rates and positive definite matrixes to make an estimation for worker service rates. To design efficient scheduling algorithms, we update dual variables by a new method, which helps us to choose a proper worker for a A-task. To analyze the proposed algorithm's performance, we define a regret (Reg) and a fit (Fit) by online convex optimization (OCO) techniques. Based on the regret and the fit, our proposed algorithms can yield a $\mathcal{O}(\sqrt{T \log T \log T})$ regret for both the single L-task case and the multi L-tasks case over T time slots.

The main contributions are shown as follows.

- We propose a new online MAB framework to address multi-dimensional large-scale tasks for maximizing total utility with the variability service of workers. In addition, this online framework can handle the short-term constraint, which yields a small constraint violation.
- We design the online algorithm to choose an appropriate worker for the total utility, which update the dual variables by a new Gradient Descent method. In addition, our efficient online algorithm is more scalable than the previous combinatorial MAB algorithms.
- We adopt a new method to analyze the online algorithm's performance which can guarantee a sublinear regret bound. In addition, we show that our designed algorithm is better than the baseline algorithms by nearly 10% for the total utility.

Organization: In Section 2, we discuss related work. In Section 3, we show system model. Section 4 presents online task assignment for a single L-task case. Section 5 discusses online scheduling for multiple L-tasks case. In Section 6, experiment is presented. Section 7 concludes this paper.

II. RELATED WORK

In crowdsourcing system, many works have studied about online task assignments for the total utility. For example, in [14], [15], [45], the utility of crowdsourcing systems is formulated by the completed tasks. In [16], the task utility is formulated by the completed tasks. To maximize the total utility, a Hungarian-based method (TGOA) is proposed by Tong in [18], which can achieve a competitive ratio of $\frac{1}{4}$ in a random model. In [17], Tong proposes a greedy-based method which is based on a Hungarian-based method and a greedy method. This method can get a competitive ratio of $\frac{1}{8}$. To maximize task utility, [18] utilizes a threshold-based method for bipartite matching and [17] utilizes the same method for trichromatic matching. If the task utility is above the beforehand threshold, this method will make a task assignment. In [19], Goel and Singla propose incentive-compatible mechanism under matching constraints, which can achieve an optimal task utility.

In addition, many research works have optimized the performance of different aspects in crowdsourcing [2], [20], [21], [39]. In particular, [20] proposes a new online cost sensitive framework to batch atomic tasks, which can reduce the cost of each atomic task. Reference [21] presents a general framework for accomplishing complex and interdependent tasks by Crowd-Forge. Crowd-Forge can decompose complex-task into small tasks, which is a prototype. Reference [2] proposes a new crowdsourcing task decomposition framework which can achieve a minimal cost by packing many atomic tasks into a big task. Reference [3] utilizes EM algorithm and Lasso regression to protect the local privacy of high-dimensional crowdsourced data.

Recently, researchers analyze the effectiveness of approximation algorithms from a bandit perspective by estimating the service rates of machines [23], [42]. The previous works explore the machine service rates in cluster. Actually, in the crowdsourcing, the service rates of workers will change in each time slot during tasks processing. Thus, we should estimate worker service rates. In this paper, we utilize the past service rates and positive definite matrixes to estimate worker service rates.

In crowdsourcing system, Multi-Armed Bandits (MAB) problem is close to our task assignment problem. Reference [22] is a combinatorial MAB problem which has studied about constrained model with a variety of budget. Moreover, this work considers the exploration of a budget-limited followed by a cost-free exploitation phase. Later, Tran-Thanh *et al.* extend the ε-first policy from [24] to an arm-limited and show that the regret of this policy is O(B2/3). Auer *et al.* show the proposed algorithm can achieve a regret with a lower bound, which is Ω(log B) regret [25]. Many works utilize worker's capacity constraint to optimize the performance of crowdsourcing platforms. Reference [18] identifies a more practical micro-task allocation problem for maximum utility in spatial crowdsourcing problem under worker's capacity constraint. Reference [31] designs a novel framework with the mutual benefit of workers, where task assignments are made under worker's capacity constraint.

In crowdsourcing system, OCO has been utilized to design the online scheduling algorithm in a convex optimization framework. Reference [26] proposes a convex optimization framework for crowds without estimating the true labels and introduces personal model of each worker. In [13], the authors consider the knapsacks problem with online convex optimization, which is stimulated by dynamic pricing in crowdsourcing. In [28], authors utilize a loss function to analyze the online learning algorithm's performance. Moreover, the loss function is a convex function which is solved by the convex optimization technique. In [27], authors propose a saddle-point modified algorithm which yields a bound of $\mathcal{O}(T^{2/3})$ for the constraint violation. Reference [30] proposes a new convex-concave approach to design the online scheduling algorithm which yields a bound of $\mathcal{O}(\sqrt{T})$ for the constraint violation.

III. PROBLEM FORMULATION AND SYSTEM MODEL

In this model, we choose workers from an available set which is a fixed set [32]. Moreover, workers are indexed from 1 to M and reach the crowdsourcing system randomly. Time is segmented into time slots. L-task j reaches the system at a_j and leaves the system at d_j [6]. A L-task (multi-dimensional large-scale task) can consist of thousands or millions of A-tasks (Atomic tasks) [2]. The L-task is a multi-dimension task specified by $\boldsymbol{v_j}$, where $\boldsymbol{v_j}$ is a m-dimensional vector, $\boldsymbol{v_j} \in [0, 1]^m$ and $\|\boldsymbol{v_j}\|_2 \leq 1$. When L-task j arrives the system, we can obtain the information of $\boldsymbol{v_j}$. In our model, each L-task allows partial execution under which all A-tasks can be processed in parallel on multiple workers. f_j is a concave utility function about X_j which is the number of completed A-tasks by deadline d_j. Fig.1 shows our crowdsourcing system model. Fig.2 shows our methodological flowchart and Table 1 presents all scheduling variables.

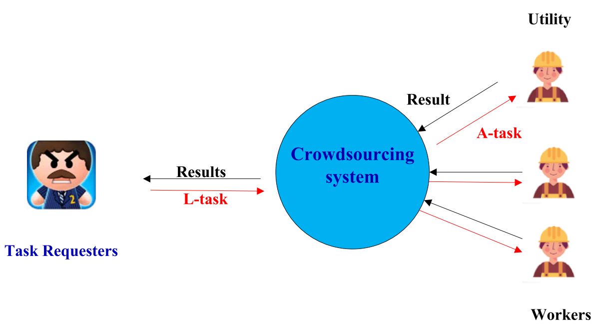


FIGURE 1. Our model.

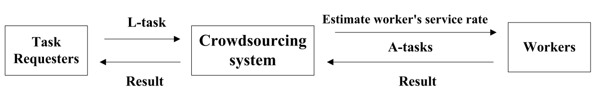


FIGURE 2. Our flowchart.

A. PROBLEM FORMULATION

To deal with the time-varying behavior of each worker, we design a stochastic model. If a worker is in specific behavior, his service rate will reduce. If a worker is working hard, his service rate will increase. In particular, we treat each varying worker as an arm for a multi-armed bandit in the system [33]. Moreover, we consider that the service rate of each arm (worker) follows a uniform random distribution in all time slots. To be more specific, we let $\boldsymbol{w_i^t}$ be the service rate of worker i in time slot t, $\boldsymbol{w_i^t} \in [0, 1]^m$ and $\|\boldsymbol{w_i}\|_2 \leq 1$.

TABLE 1. Scheduling variables.

Variables	Mean
a_j	the arrival time L-task j
d_j	the leaving time L-task j
f_j	the utility of L-task j
M	workers' number
N	L-tasks' number
$\boldsymbol{v_j}$	the feature vector of L-task j
$\boldsymbol{w_i^t}$	the varying service of worker i at slot t
r_i	the consume resource of worker i at slot t
C_i	the capacity of worker i
$x_j^i(t)$	a scheduling decision variable
$Reg(T)$	a regret of our algorithm at the end
$Fit(T)$	a fit of our algorithm at the end
$g(x(t))$	the short-term constraint of our model
M_t	a positive definite matrix in time slot t
η_τ	a real number in time slot τ
I	a positive definite matrix
τ	a time slot
β	a step-size
γ	a step-size
z_τ	a real number in slot τ
α	a dual variable in our model
L_t	the Lagrangian function in our model

In this model, $\boldsymbol{w_i^t}$ follows a uniform random distribution in all time slots with $\mathbb{E}[\boldsymbol{w_i^t}] = \boldsymbol{w_i}$ for all i. In addition, worker i will consume resource r_i to process each A-task. In each time slot, the total resource consumption should meet worker's capacity C_i. $x_j^i(t)$ is considered as a scheduling variable which is a binary variable to indicate whether worker i has processed L-task j at time slot t. Then, the completed A-tasks of L-task $X_j(t)$ is represented by:

$$X_j(t) = \sum_{i=1}^{M} [\boldsymbol{w_i^t}]^\top \cdot x_j^i(t) \cdot \boldsymbol{v_j} \,, \tag{1}$$

Thus, X_j can be given by:

$$X_j = \sum_{t=a_j}^{d_j} \sum_{i=1}^{M} [\boldsymbol{w_i^t}]^\top \cdot x_j^i(t) \cdot \boldsymbol{v_j} \,. \tag{2}$$

In this model, we maximize the total utility by scheduling variable $\boldsymbol{x}(t)$ which should satisfy resource constraint. Thus, in crowdsourcing system, the utility of optimization problem P1 is given by:

$$\max_{\{\boldsymbol{x}(t)\}} \sum_{j=1}^{N} f_j\Big(\sum_{t=a_j}^{d_j} \sum_{i=1}^{M} [\boldsymbol{w_i^t}]^\top \cdot x_j^i(t) \cdot \boldsymbol{v_j} \Big) \tag{P1}$$

$$\text{s.t.} \sum_{j=1}^{N} x_j^i(t) \cdot r_i \leq C_i \quad \forall i, \tag{3}$$

$$x_j^i(t) \in \{0, 1\} \,, \quad \forall j, i, t. \tag{4}$$

Constraint (3) states that the total resource consumption of A-tasks should satisfy worker's capacity C_i in time slot t. In P1 problem, for L-task j, $1 \leq i \leq M$, $1 \leq j \leq N$ and $a_j \leq t \leq d_j$. P1 is a combinatorial MAB problem to maximize the total utility, which chooses a subset of proper arms (workers) in each time slot.

B. MAXIMIZE L-TASKS UTILITY

In optimization problem P1, we can only quantify the utility of a L-task after this L-task is completed and leaves the system. Thus, in each time slot, we should make the online decisions for task assignments before this L-task leaves the system. As such, it is not tractable to tackle problem P1 since the aim cannot be handled in an online manner. In this paper, we utilize another approximated optimization problem to instead optimization problem P1. Then, we can solve the approximated optimization problem to get online algorithms. This approximated optimization problem is also a time-varying aim problem as follow:

$$\sum_{j=1}^{N}(d_j-a_j)f_j(\sum_{t=a_j}^{d_j}\sum_{i=1}^{M}x_j^i(t)[\boldsymbol{w_i^t}]\top\cdot\boldsymbol{v_j}/(d_j-a_j))\ . \tag{5}$$

(d_j-a_j) is the processing time of L-task j, which is also the number of time slots in crowdsourcing system. In each time slot, $f_j(\sum_{t=a_j}^{d_j}\sum_{i=1}^{M}[\boldsymbol{w_i^t}]\top\cdot x_j^i(t)\cdot\boldsymbol{v_j}/(d_j-a_j))$ is average utility and we should maximize this average utility. We define the regret and the fit by online convex optimization (OCO) techniques, which are the performance metrics of our proposed online algorithm [23], [34].

C. PERFORMANCE METRICS

We use the regret as a performance metrics to analyze our online algorithm. This regret analysis is better than a competitive-ratio analysis which is not always doable to achieve a constant ratio. In this model, the optimal utility of P1 is $f_j(\sum_{t=a_j}^{d_j}\sum_{i=1}^{M}[\boldsymbol{w_i^t}]\top\cdot x_j^{i,*}(t)\cdot\boldsymbol{v_j})$ and the average utility in Eq.(5) is $(d_j-a_j)f_j(\sum_{t=a_j}^{d_j}\sum_{i=1}^{M}[\boldsymbol{w_i^t}]\top\cdot x_j^i(t)\cdot\boldsymbol{v_j}/(d_j-a_j))$. Based on [35], we define the regret is the difference between the optimal utility and the average utility.

$$\begin{aligned}\text{Reg}(T)=&\sum_{j=1}^{N}f_j(\sum_{t=a_j}^{d_j}\sum_{i=1}^{M}[\boldsymbol{w_i^t}]\top\cdot x_j^{i,*}(t)\cdot\boldsymbol{v_j})\\&-\sum_{j=1}^{N}(d_j-a_j)f_j(\sum_{t=a_j}^{d_j}\sum_{i=1}^{M}[\boldsymbol{w_i^t}]\top x_j^i(t)\boldsymbol{v_j}/(d_j-a_j))\ .\end{aligned} \tag{6}$$

where $x_j^{i,*}(t)$ is optimal decision for P1. Essentially, we use the optimal utility as the benchmark to compare the average utility. Moreover, our online solutions meet the short-term constraint as follows:

$$\text{Fit}(T)=\sum_{i=1}^{M}\sum_{t=a_j}^{d_j}(\sum_{j=1}^{N}x_j^i(t)\cdot r_i-C_i)\ , \tag{7}$$

In some time slots, Eq.(7) can be violated temporarily [30]. Moreover, $(C_i-\sum_{j=1}^{N}x_j^i(t)\cdot r_i)$ is considered as the constraint violation, which is under control in our model.

IV. A SINGLE L-TASK CASE ONLINE TASK ASSIGNMENT

In our model, we first design online task assignment policy in a single L-task case. In optimization problem P1, variable $x_j^i(t)$ becomes $x^i(t)$ and f_j becomes f. We handle this case by the OCO techniques and consider $a_j=0, d_j=T$. In each time slot, the short-term constraint in (3) can be given by:

$$g(\boldsymbol{x}(t))=x^i(t)\cdot r_i-C_i\leq 0\ , \tag{8}$$

Combine Eq.(5) with Eq.(8), P1 becomes:

$$\max_{\{\boldsymbol{x}(t)\}}\ T\cdot f(\sum_{t=1}^{T}\sum_{i=1}^{M}[\boldsymbol{w_i^t}]\top\cdot x_j^i(t)\cdot\boldsymbol{v_j}/T) \tag{P2}$$

$$\text{s.t. } g(x(t))\leq 0\ . \tag{9}$$

A. ESTIMATE WORKER'S SERVICE RATE

In P2 problem, we cannot know the service rate $\boldsymbol{w_i^t}$ at the beginning of slot t. Moreover, $\boldsymbol{w_i^t}$ will change by each time slot t. Thus, we need to estimate the service rate $\widehat{\boldsymbol{w_i^t}}$ before making online task assignments. We use the past service rate of arm (worker) i and positive definite matrixes to estimate the service rate $\boldsymbol{w_i^t}$. This technique is common in prior work on contextual bandits (e.g., in [29], [36], [37], [43]). By the multi-dimension task method of [43], $\widehat{\boldsymbol{w_i^t}}$ is given by:

$$\widehat{\boldsymbol{w_i^t}}=M_t^{-1}\sum_{\tau=1}^{t-1}x^i(\tau)\boldsymbol{v_j}z_\tau\ , \tag{10}$$

where

$$\text{M}_t=I+\sum_{\tau=1}^{t-1}x^i(\tau)\boldsymbol{v_j}[x^i(\tau)\boldsymbol{v_j}]^\top,\ \ z_\tau=[\widehat{\boldsymbol{w_i^\tau}}]^\top x^i(\tau)\boldsymbol{v_j}+\eta_\tau\ , \tag{11}$$

where M_t is the positive definite matrixe in time slot t and I is also a positive definite matrixe. z_τ and $\eta_\tau\in\mathbb{R}$, $\mathbb{E}[\eta_\tau|x_j^i(1)\boldsymbol{v_j},z_1,\ldots,x_j^i(\tau-1)\boldsymbol{v_j},z_{\tau-1},x_j^i(\tau)\boldsymbol{v_j}]=0$. Moreover, t and τ denote time slots.

B. DESIGN ONLINE ALGORITHM

In this section, we utilize random sampling to update the primal-dual variables based on a primal-dual approach [38]. We consider that $\alpha(t+1)$ is the dual variable. Therefore, in time slot t, the Lagrangian function is characterized by:

$$\begin{aligned}L_t(\boldsymbol{x},\boldsymbol{\alpha})=tf((&\sum_{\tau=1}^{t-1}\sum_{i=1}^{M}[\widehat{\boldsymbol{w_i^\tau}}]\top x^i(\tau)\boldsymbol{v_j}+\sum_{i=1}^{M}[\widehat{\boldsymbol{w_i^t}}]\top x^i(t)\boldsymbol{v_j})/t)\\&-\alpha(t+1)g(\boldsymbol{x})\ .\end{aligned} \tag{12}$$

By the primal-dual approach in [38], $\boldsymbol{x}(t+1)$ is given by:

$$\boldsymbol{x}(t+1)=\Pi_\Omega\Big(\boldsymbol{x}(t)+\beta\cdot\nabla_{\boldsymbol{x}}L_t(\boldsymbol{x}(t),\boldsymbol{\alpha})\Big)\ , \tag{13}$$

Here, $\Pi_\Omega(\boldsymbol{n})$ is the projection of $\boldsymbol{n}$ and β is the step-size.

As we know, $\Pi_\Omega(\boldsymbol{n})$ is the vector with all values between 0 and 1. Thus, $\Pi_\Omega(\boldsymbol{n})$ can be calculated by:

$$(\Pi_\Omega(\boldsymbol{n}))^i=\begin{cases}n^i & 0\leq n^i\leq 1\ ,\\1 & n^i>1\ ,\\0 & \text{otherwise}\ ,\end{cases} \tag{14}$$

where n^i is the one element in vector $\Pi_{\Omega}(\boldsymbol{n})$.

We update the dual variable by Nesterov's Accelerated Gradient Descent [40]. For the reason that this method can achieve a better regret bound than others. Thus, the dual variable is given by:

$$\alpha(t+1) = \max \left\{0\,, \alpha(t) + 2\mu \cdot g(\boldsymbol{x}(t)) - \mu \cdot g(\boldsymbol{x}(t-1))\right\}\,. \tag{15}$$

Here μ is the step size and $g(\boldsymbol{x}(t))$ is the short-term constraint in our model.

In addition, $x_j^i(t)$ is considered as a scheduling variable, which is a binary variable to indicate whether worker i has processed L-task j at time t. Based on Eq.(13), we can know that $\boldsymbol{x}(t)$ is the vector with all values between 0 and 1. Thus, we utilize a simple random sampling method to round $x^i(t)$ as an integer $\widetilde{x}^i(t)$:

$$\widetilde{x}^i(t) = \begin{cases} 1 & \text{with prob. } x^i(t)\,, \\ 0 & \text{with prob. } (1 - x^i(t))\,. \end{cases} \tag{16}$$

Furthermore, we can get:

$$\mathbb{E}[\widetilde{x}^i(t)] = x^i(t)\,. \tag{17}$$

where $\widetilde{x}^i(t)$ is considered as the unbiased estimator of $x^i(t)$.

To solve the optimization problem P2, we first utilize Eq.(10) to estimate the service rate $\widehat{\boldsymbol{w}_i^t}$. Then, we use Eq.(15) to calculate the dual variable $\alpha(t+1)$. At last, we substitute $\widehat{\boldsymbol{w}_i^t}$ and $\alpha(t+1)$ into Lagrangian function Eq.(12) to get the primal variable $\boldsymbol{x}(t)$. Thus, we round the decimal $x^i(t)$ to an integer $\widetilde{x}^i(t)$ by Eq.(16). Then, we check whether the constraints $\sum_{j=1}^{N} \widetilde{\boldsymbol{x}}(t) \cdot r_i \leq C_i$ & $\widetilde{x}^i(t) == 1$ are satisfied. If these conditions are satisfied, we assign L-Task j to worker i. The corresponding pseudo-code of *OSS* (Online Algorithm by Random Sampling in A Single L-Task) is given in Algorithm 1.

Algorithm 1 :OSS

```
1: Initialization α(0) = 0 and x̃^i(t) = 1;
2: for t = 1, 2, ..., T do
3:     Estimation on worker service rate ŵ_i^t by Eq.(10);
4:     Calculate the dual variable α(t + 1) by Eq.(15);
5:     Calculate the primal variable x(t) by Eq.(13);
6:     Round x(t) to integer x̃(t) by Eq.(16);
7:     if x̃^i(t) · r_i ≤ C_i & x̃^i(t) == 1 then
8:         Assign L-task j to worker i;
9:     else
10:        exit;
11:    end if
12: end for
```

C. OSS ALGORITHM'S PERFORMANCE

In our model, we analyze OSS algorithm's performance using regret metrics defined in Eq.(7) and Eq.(6). If the OSS algorithm guarantees a sublinear regret bound and the constraint violations, our online solutions (task assignments) are close to the optimal solutions over time slots. Our probabilistic model takes into account the random behavior of workers in crowdsourcing systems, which can apply for real applications.

Theorem 1: After T time slots, the upper bound of Fit(T) in Eq.(7) is:

$$Fit(T) \leq \sqrt{TMC_i^{max} \ln \frac{T}{\delta}} + \ln \frac{T}{\delta}\,, \tag{18}$$

with prob. $(1-\delta)$.

Proof: To prove this theorem, based on Eq.(17), we can get:

$$\mathbb{E}[\widetilde{x}^i(t)] = x^i(t), \quad t \in \{1, 2, \dots, T\}\,.$$

Considering Eq.(8) and OSS algorithm, in a single L-Task case, Fit(T) is:

$$\text{Fit}(T) = \sum_{i=1}^{M}\sum_{t=1}^{T}(r_i \cdot \widetilde{x}^i(t) - C_i)\,, \tag{19}$$

We let $B = \sum_{t=1}^{T}\sum_{i=1}^{M} r_i \cdot x_j^i(t)$ and $\widetilde{B} = \sum_{t=1}^{T}\sum_{i=1}^{M} r_i \cdot \widetilde{x}^i(t)$. Then, [41] implies that, at least $(1-\delta)$ probability, we can get:

$$|B/T - \widetilde{B}/T| \leq \sqrt{\frac{\gamma \widetilde{B}}{T^2} + \frac{\gamma}{T}}\,, \tag{20}$$

which leads to

$$\widetilde{B} - B \leq \sqrt{\gamma \widetilde{B} + \gamma}\,, \tag{21}$$

In our model, the constraint (8) implies that $B \leq \sum_{i=1}^{M}\sum_{t=1}^{T} C_i$. Therefore, we have:

$$\text{Fit}(T) \leq \sqrt{\gamma TMC_i^{max}} + \gamma\,. \tag{22}$$

holds with prob. $(1-\delta)$.

Let $\gamma = \ln \frac{T}{\delta}$. Thus, we can get:

$$\text{Fit}(T) \leq \sqrt{TMC_i^{max} \ln \frac{T}{\delta}} + \ln \frac{T}{\delta}\,, \tag{23}$$

□

Theorem 2: After T time slots, if f is a π-Lipschitz function, the upper bound of Reg(T) in Eq.(6) is:

$$Reg(T) \leq \mathcal{O}(\pi 2Mm\sqrt{T \ln \frac{1+Tm}{\delta} \ln T}) + 2\pi \ln \frac{T}{\delta}\,. \tag{24}$$

with prob. $(1-\delta)$.

Lemma 1: The $\mathbf{x}(t+1)$ *in Eq.*(13) *is calculated by:*

$$\boldsymbol{x}(t+1) = \arg\min_{\boldsymbol{x}\in\Omega} -\boldsymbol{\sigma}_t^T(\mathbf{x} - \boldsymbol{x}(t)) + \frac{||\boldsymbol{x} - \boldsymbol{x}(t)||_2^2}{2\beta}\,, \tag{25}$$

where $\boldsymbol{\sigma}_t = \beta \cdot \nabla_{\boldsymbol{x}} L_t(\boldsymbol{x}(t), \alpha(t+1))$.

Appendix VII-A is the proof.

Lemma 2: With prob. $(1-\delta)$, *we have*

$$f(\sum_{t=1}^{T}\sum_{i=1}^{M}[\boldsymbol{w}_i^t]^\top \cdot x^{i,*}(t) \cdot \boldsymbol{v}_j) \leq \sum_{t=1}^{T} f(\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^t}]^\top \hat{x}^{i,*}\boldsymbol{v}_j)\,. \tag{26}$$

Appendix VII-B is the proof.

Lemma 3: Let $\beta = \frac{1}{2\sqrt{T}}$*, we have:*

$$\sum_{t=1}^{T} f(\sum_{i=1}^{M}[\widehat{\mathbf{w}_i^t}]^\top \hat{x}^{i,*}(t)\mathbf{v}_j) - T\cdot f(\sum_{t=1}^{T}\sum_{i=1}^{M}[\widehat{\mathbf{w}_i^t}]^\top x^i(t)\mathbf{v}_j/T) \le \frac{\sqrt{T}MD^2}{2} + \frac{3M\sqrt{T}}{2}. \quad (27)$$

Appendix VII-C is the proof.

Proof: As we know, $\widehat{\mathbf{w}_i^t}$ is the estimation of service rate, which is not the real service rate. Thus, we can get:

$$\mathrm{Reg}(T) = f_j(\sum_{t=1}^{T}\sum_{i=1}^{M}[\mathbf{w}_i^t]^\top \cdot x_j^{i,*}(t)\cdot \mathbf{v}_j) - Tf(\sum_{t=1}^{T}\sum_{i=1}^{M}[\mathbf{w}_i^t]^\top \widetilde{x}^i(t)\mathbf{v}_j/T) \quad (28)$$

Based on Lemma 2 and Lemma 3, we can use the intermediate variables $f(\sum_{t=1}^{T}\sum_{i=1}^{M}[\widehat{\mathbf{w}_i^t}]^\top x^i(t)\mathbf{v}_j/T)$ to get the regret. In the following, we show the relationship between $f(\sum_{t=1}^{T}\sum_{i=1}^{M}[\widehat{\mathbf{w}_i^t}]^\top x^i(t)\mathbf{v}_j/T)$ and $f(\sum_{t=1}^{T}\sum_{i=1}^{M}[\mathbf{w}_i^t]^\top \widetilde{x}^i(t)\mathbf{v}_j/T)$. If f is a π-Lipschitz function, we can get:

$$\begin{aligned}
&\Big|Tf(\sum_{t=1}^{T}\sum_{i=1}^{M}[\widehat{\mathbf{w}_i^t}]^\top x^i(t)\mathbf{v}_j/T) - Tf(\sum_{t=1}^{T}\sum_{i=1}^{M}[\mathbf{w}_i^t]^\top \widetilde{x}^i(t)\mathbf{v}_j/T)\Big| \\
&\le \pi\cdot\Big|\sum_{t=1}^{T}\sum_{i=1}^{M}[\widehat{\mathbf{w}_i^t}]^\top x^i(t)\mathbf{v}_j - \sum_{t=1}^{T}\sum_{i=1}^{M}[\mathbf{w}_i^t]^\top \widetilde{x}^i(t)\mathbf{v}_j\Big| \\
&\le \pi\Big|\underbrace{\sum_{t=1}^{T}\sum_{i=1}^{M}[\widehat{\mathbf{w}_i^t}]^\top x^i(t)\mathbf{v}_j}_{R_1} - \underbrace{\sum_{t=1}^{T}\sum_{i=1}^{M}[\widehat{\mathbf{w}_i^t}]^\top \widetilde{x}^i(t)\mathbf{v}_j}_{R_2}\Big| \\
&+\pi\Big|\sum_{t=1}^{T}\sum_{i=1}^{M}[\widehat{\mathbf{w}_i^t}]^\top \widetilde{x}^i(t)\mathbf{v}_j - \underbrace{\sum_{t=1}^{T}\sum_{i=1}^{M}[\mathbf{w}_i^t]^\top \widetilde{x}^i(t)\mathbf{v}_j}_{R_3}\Big|. \quad (29)
\end{aligned}$$

Based on Eq.(20) and $\gamma = \ln\frac{T}{\delta}$, with prob. $(1-\delta)$, we can get to know that:

$$|R_1 - R_2| \le \sqrt{\ln\frac{T}{\delta}TM} + \ln\frac{T}{\delta}. \quad (30)$$

The second term of Eq.(29) can be reformulated as:

$$|R_2 - R_3| = \Big|\sum_{t=1}^{T}\sum_{i=1}^{M}[\widehat{\mathbf{w}_i^t}]^\top \widetilde{x}^i(t)\mathbf{v}_j - \sum_{t=1}^{T}\sum_{i=1}^{M}[\mathbf{w}_i^t]^\top \widetilde{x}^i(t)\mathbf{v}_j\Big|. \quad (31)$$

Following [37], $\mathbf{w}_i^t$ is in an ellipsoid with the center $\widehat{\mathbf{w}_i^t}$ at least $(1-\delta)$ probability. In this situation, we define the positive definite matrixe M-norm as $\|\mathbf{w}_i^t\|_M := \sqrt{[\mathbf{w}_i^t]^\top M \mathbf{w}_i^t}$. In each time slot, the confidence ellipsoid is formulated by:

$$\|\widehat{\mathbf{w}_i^t} - \mathbf{w}_i^t\|_{M_t} \le \sqrt{m\ln\frac{1+tm}{\delta}} + \sqrt{m} \quad (32)$$

Then, lemma 11 of [36] implies that $\sum_{t=1}^{T}\|\widetilde{x}^i(\tau)\mathbf{v}_j\|_{M_t^{-1}} \le \sqrt{mT\ln T}$. Thus, with probability at least $(1-\delta)$, we have

$$\begin{aligned}
&\pi\Big|\sum_{t=1}^{T}\sum_{i=1}^{M}[\widehat{\mathbf{w}_i^t}]^\top \widetilde{x}^i(t)\mathbf{v}_j - \sum_{t=1}^{T}\sum_{i=1}^{M}[\mathbf{w}_i^t]^\top \widetilde{x}^i(t)\mathbf{v}_j\Big| \\
&\le \pi M\|\mathbf{w}_i^t - \widehat{\mathbf{w}_i^t}\|_{M_t}\sum_{t=1}^{T}\|\widetilde{x}^i(t)\mathbf{v}_j\|_{M_t^{-1}} \\
&\le \pi M(\sqrt{m\ln\frac{1+tm}{\delta}} + \sqrt{m})(\sqrt{mT\ln T}) \\
&\le \pi 2Mm\sqrt{T\ln\frac{1+Tm}{\delta}\ln T} \quad (33)
\end{aligned}$$

Combining Eq.(29), Eq.(30) and Eq.(33), with prob. $(1-\delta)$, we have:

$$\begin{aligned}
&\Big|Tf(\sum_{t=1}^{T}\sum_{i=1}^{M}[\widehat{\mathbf{w}_i^t}]^\top x^i(t)\mathbf{v}_j/T) - Tf(\sum_{t=1}^{T}\sum_{i=1}^{M}[\mathbf{w}_i^t]^\top \widetilde{x}^i(t)\mathbf{v}_j/T)\Big| \\
&\le \mathcal{O}(\pi 2Mm\sqrt{T\ln\frac{1+Tm}{\delta}\ln T}) + 2\pi\ln\frac{T}{\delta}. \quad (34)
\end{aligned}$$

Together Eq. (34) with results from Lemma 2 and Lemma 3, we can get Theorem 2. □

V. ONLINE SCHEDULING FOR MULTIPLE L-TASKS CASE

In this section, we show online task assignments in multiple L-tasks case. Based on Eq.(8), the capacity constraint can be characterized by:

$$g(\mathbf{x}_j(t)) = \sum_{j=1}^{N} x_j^i(t)\cdot r_i - C_i \le 0. \quad (35)$$

In this multiple L-Tasks case, we let $a_j = 0$ and $d_j = T$. Then, we present the optimization problem as follow:

$$\max_{\{\mathbf{x}(t)\}} \sum_{j=1}^{N}\sum_{t=1}^{T} t\cdot f_j\Big(\frac{\sum_{\tau=1}^{t}\sum_{i=1}^{M}[\mathbf{w}_i^\tau]^\top x_j^i(\tau)\mathbf{v}_j}{t}\Big) \quad (P3)$$

$$\text{s.t. } g(\mathbf{x}_j(t)) \le 0, \quad \forall j, \quad (36)$$

A. DESIGN ONLINE ALGORITHM

In P3 problem, the service rate $\mathbf{w}_i^t$ is also unknown. Thus, we utilize the previous service rate of arm (worker) and positive definite matrixes to estimate the service rate $\mathbf{w}_i^t$. Thus, $\widehat{\mathbf{w}_i^t}$ is given by:

$$\widehat{\mathbf{w}_i^t} = M_t^{-1}\sum_{\tau=1}^{t-1}\sum_{j=1}^{N} x^i(\tau)\mathbf{v}_j z_\tau, \quad (37)$$

where

$$\mathrm{M}_t = I + \sum_{\tau=1}^{t-1} x^i(\tau)\mathbf{v}_j[x^i(\tau)\mathbf{v}_j]^\top, \quad z_\tau = [\widehat{\mathbf{w}_i^\tau}]^\top x^i(\tau)\mathbf{v}_j + \eta_\tau, \quad (38)$$

where M_t and I are positive definite matrixes, $\eta_\tau \in \mathbb{R}$ and $\mathbb{E}[\eta_\tau | x_j^i(1)\mathbf{v}_j, z_1, \ldots, x_j^i(\tau-1)\mathbf{v}_j, z_{\tau-1}, x_j^i(\tau)\mathbf{v}_j] = 0$.

We consider a dual variable $\alpha_j(t)$ and $\boldsymbol{\alpha}(t) = \{\alpha_1(t), \ldots, \alpha_N(t)\}$. In each time slot, the Lagrangian function can be formulated as:

$$\begin{aligned} &L_t(\boldsymbol{x}, \boldsymbol{\alpha}(t+1)) \\ &= \sum_{j=1}^{N} tf_j\Big(\frac{\sum_{\tau=1}^{t-1}\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^{\tau}}]\top x_j^i(\tau)\boldsymbol{v_j} + \sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^{t}})]\top x_j^i \boldsymbol{v_j}}{t}\Big) \\ &\quad - \sum_{j=1}^{N} \alpha_j(t+1) g(\boldsymbol{x}^i(t)) \,. \end{aligned} \tag{39}$$

In this case, multiple L-Tasks $\boldsymbol{J}_t$ may arrive at the same worker simultaneously. We utilize the same method of Eq. (13) to get $\boldsymbol{x}(t)$ as follow:

$$\boldsymbol{x}(t) = \Pi_\Omega\Big(\boldsymbol{x}(t-1) + \beta \cdot \nabla_{\boldsymbol{x}} L_{t-1}(\boldsymbol{x}(t-1), \boldsymbol{\alpha})\Big) \tag{40}$$

Then, we should schoose a proper L-Task among $\boldsymbol{J}_t$. The proper L-Task can be choosen by $j = \mathbf{min}_{j \in \boldsymbol{J}_t} \big\{\boldsymbol{x}_j(t) - (\boldsymbol{x}_j(t-1) + \beta \cdot \nabla_{\boldsymbol{x}_j} L_{t-1}(\boldsymbol{x}(t-1), \boldsymbol{\alpha}(t)))\big\}$.

After getting $x_j^i(t)$, we round $x_j^i(t)$ to an integer $\widetilde{x}_j^i(t) \in \{0, 1\}$ by the method in Eq.(16). Thus, $\widetilde{x}_j^i(t)$ meets:

$$\mathbb{E}[\widetilde{x}_j^i(t)] = x_j^i(t) \,. \tag{41}$$

We update $\alpha_j(t+1)$ by the method in Eq.(15) as follows:

$$\alpha_j(t+1) = \max \Big\{0\,, \alpha_j(t) + 2\mu \cdot g(\boldsymbol{x}_j(t)) - \mu \cdot g(\boldsymbol{x}_j(t-1))\Big\} \,. \tag{42}$$

The corresponding pseudo-code of *OSM* (Online Algorithm by Random Sampling in Multiple L-Tasks) is presented in Algorithm 2.

Algorithm 2 :OSM

```
for t = 1, 2, ..., T do
    Initialization α(0) = 0 and x̃_j^i(t) = 1;
    Estimate worker service rate ŵ_i^t by Eq.(37) Eq.(38);
    Calculate x_j(t) by Eq.(40);
    for i = 1, 2, ..., M do
        Choose the proper L-Task j by the upper method;
        Calculate the dual variable α_j(t + 1) by Eq.(42);
        Round x_j^i(t) to x̃_j^i(t) by Eq.(16);
        if Σ_{j=1}^N x̃_j^i(t) · r_i ≤ C_i & x̃_j^i(t) == 1 then
            Assign L-task j to worker i;
        else
            exit;
        end if
    end for
end for
```

B. OSM ALGORITHM'S PERFORMANCE

In our model, we analyze OSM algorithm's performance using regret metrics. If the OSM algorithm guarantees a sub-linear regret bound and the constraint violations, our online solutions (task assignments) are close to the optimal solutions over time slots.

Theorem 3: After T time slots, the upper bound of Fit(T) in OSM is:

$$Fit(T) \leq \sqrt{TMC_i^{max} \ln \frac{T}{\delta}} + \ln \frac{T}{\delta} \,, \tag{43}$$

with prob. $(1-\delta)$.

Proof: Inferred from Eq. (20) and the constraint (35), we let $B = \sum_{j=1}^{N}\sum_{t=1}^{T}\sum_{i=1}^{M} r_i \cdot x_j^i(t)$ and $\widetilde{B} = \sum_{j=1}^{N}\sum_{t=1}^{T}\sum_{i=1}^{M} r_i \cdot \widetilde{x}^i(t)$. Fit($T$) in Eq.(7) under OSM is bounded by:

$$\text{Fit}(T) \leq \sqrt{TMC_i^{max} \ln \frac{T}{\delta}} + \ln \frac{T}{\delta} \,. \tag{44}$$

□

Theorem 4: After T time slots, if f is a π-Lipschitz function, the upper bound of Reg(T) under OSM is:

$$Reg(T) \leq \mathcal{O}(\pi 2NMm\sqrt{T \ln \frac{1+Tm}{\delta} \ln T}) + 2\pi N \ln \frac{MT}{\delta} \,. \tag{45}$$

Proof: As f is a π-Lipschitz function, following Eq.(34), we can get that:

$$\begin{aligned} &\Big|Tf(\sum_{t=1}^{T}\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^t}]\top x_j^i(t)\boldsymbol{v_j}/T) - Tf(\sum_{t=1}^{T}\sum_{i=1}^{M}[\boldsymbol{w}_i^t]\top \widetilde{x}_j^i(t)\boldsymbol{v_j}/T)\Big| \\ &\quad \leq \mathcal{O}(\pi 2Mm\sqrt{T \ln \frac{1+Tm}{\delta} \ln T}) + 2\pi \ln \frac{T}{\delta} \,. \end{aligned} \tag{46}$$

Thus, with prob. $(1-\delta)$, we can get:

$$\begin{aligned} &\Big|\sum_{j=1}^{N} Tf(\sum_{t=1}^{T}\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^t}]\top x_j^i(t)\boldsymbol{v_j}/T) \\ &\quad - \sum_{j=1}^{N} Tf(\sum_{t=1}^{T}\sum_{i=1}^{M}[\boldsymbol{w}_i^t]\top \widetilde{x}_j^i(t)\boldsymbol{v_j}/T)\Big| \\ &\leq \sum_{j=1}^{N}\Big|Tf(\sum_{t=1}^{T}\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^t}]\top x_j^i(t)\boldsymbol{v_j}/T) \\ &\quad - Tf(\sum_{t=1}^{T}\sum_{i=1}^{M}[\boldsymbol{w}_i^t]\top \widetilde{x}_j^i(t)\boldsymbol{v_j}/T)\Big| \\ &\leq \mathcal{O}(N\pi 2Mm\sqrt{T \ln \frac{1+Tm}{\delta} \ln T}) + N2\pi \ln \frac{T}{\delta} \,. \end{aligned} \tag{47}$$

Combine Eq. (47) with Lemma 2 and Lemma 3, we can get Theorem 4. □

VI. EXPERIMENT

In this section, we investigate online algorithm's performance using a real data set. In this data set, we can get the duration of all L-tasks and the number of A-tasks. Moreover, we extract more than 6000 L-tasks over 12 hours. $\boldsymbol{v_j}$ is 5-dimensional vector, $\boldsymbol{v_j} \in [0, 1]^5$ and $\|\boldsymbol{v_j}\|_2 \leq 1$. $\boldsymbol{w}_i^t$ is 5-dimensional vector, $\boldsymbol{w}_i^t \in [0, 1]^5$ and $\|\boldsymbol{w}_i^t\|_2 \leq 1$. r_i is a value and $r_i \in [0, 1]$. The detailed L-task datas are illustrated in Table 2.

TABLE 2. Real data.

Total number of all L-tasks	6060
Trace duration of all L-tasks (s)	35030
Average number of A-tasks	25.31
Minimum duration (s)	10.8
Maximum duration (s)	22920.3
Average duration (s)	1180.7

In practical applications, translating Chinese papers into English papers are typical L-tasks, which need many workers to complete. As we know, Google Translating Software is not suitable for completing these L-tasks, since it can lead to many incorrect sentences. Thus, we should upload these L-tasks into Amazon Mechanical Turk in turn. Each paper contains many sentences which can be assigned to many workers simultaneously. Each worker translates one sentence on three aspects including context, grammar and spelling. At the beginning of each time slot, we should estimate the service rate $\boldsymbol{w}_i^t$. If one worker makes a better translation, he or she will get more payment.

Simulation Setup: In the real data, we can get the arrival time a_j and the deadline d_j. The capacity C_i is uniformly distributed in [5,10]. Based on [44], a concave function $f_j(X_j) = v_j X_j^{\kappa}$ can be considered as the utility function of L-task j. Here, we let $\kappa = 1/2$, v_j is also a uniform distribution in [1,5].

A. EXPERIMENT FOR OSM

Baseline Algorithms: The following three algorithms are considered as the baseline algorithms to compare with the OSM algorithm:

- *OSM No Worker Estimations (OSMN):* A L-task is assigned to a worker by OSM algorithm, but this algorithm does not estimate the worker service variabilities.
- *Random Algorithm (RA):* A L-task is assigned to a worker randomly.
- *Deadline-Aware Algorithm (DA):* A L-task is assigned to a worker by the tighter deadline.

In OSM algorithm, in each time slot, we maximize the average utility by assigning a task to an appropriate worker. Therefore, we can know that the number of workers can affect the total utility in crowdsourcing system. Results in Fig. 3 present the total utility of 200 workers is more than the total utility of others. Considering the total utility and the economic efficiency, we let the number of workers be 200 and neglect more than 1600 situations in OSM algorithm.

In OSM algorithm, different time slots can include different number of L-tasks, which can affect the remaining task assignments. Therefore, we need to calculate out the total utility by different time slots. Fig. 4 presents the total utility of 5 seconds is more than the total utility of other time slots. Considering the total utility and the completing time, we let time slot be 5 seconds and neglect more than 40 seconds situations in OSM algorithm.

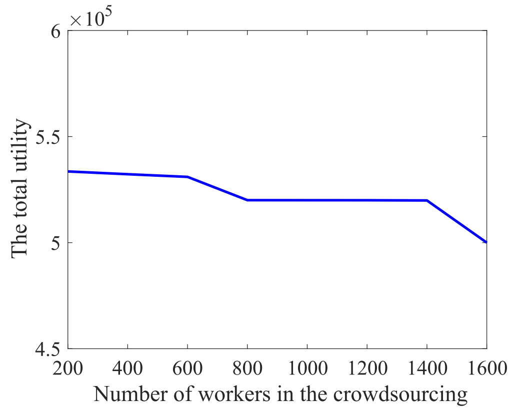


FIGURE 3. The total utility by different number of workers.

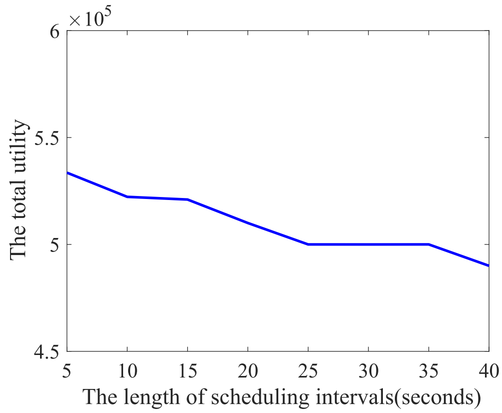


FIGURE 4. The total utility by different time slots.

By 5 seconds and 200 workers, we compare the total utility of OSM algorithm with that of OSMN algorithm, RA algorithm and DA algorithm. Fig. 5 shows the total utility curves of four algorithms. The curve of OSM algorithm is above that of OSMN algorithm, RA algorithm and DA algorithm. As we know, OSM algorithm maximizes the total utility by Online Convex Optimization (OCO) techniques, which is better than other algorithms.

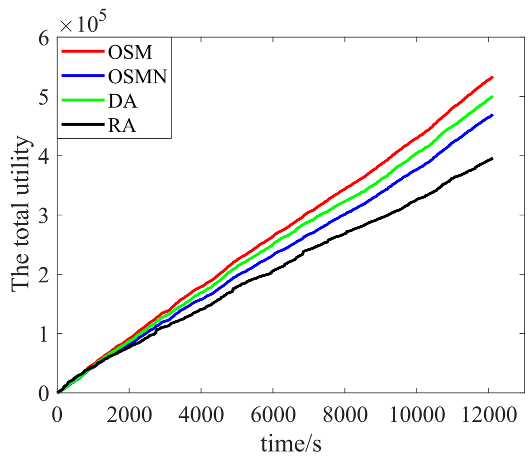


FIGURE 5. The total utility of all L-tasks.

TABLE 3. The total utility by different algorithms.

OSM Algorithm	5.33×10^5
OSM No estimation Algorithm	4.69×10^5
Random Algorithm	3.96×10^5
Deadline-aware Algorithm	5.00×10^5

In Table 3, we calculate out the total utility by four algorithms in crowdsourcing system. We can know that, comparing to OSMN algorithm, RA algorithm and DA algorithm, OSM algorithm increases the total utility by 15%, 25% and 5% respectively. Thus, OSM algorithm presents a better performance than other three baseline algorithms.

VII. CONCLUSION

In this paper, we address large-scale multi-dimensional crowdsourcing tasks by considering the varying service of workers in crowdsourcing system. We consider each varying worker as an arm for a multi-armed bandit in the system. In this model, we study the combinatorial MAB problem with concave aim. To tackle this challenge, we have presented a novel framework with bandit method. We design the online scheduling algorithm from a bandit perspective by Online Convex Optimization (OCO) techniques. Our algorithms can achieve a sublinear regret bound and show a better performance than the baseline algorithms.

APPENDIX

A. PROOF OF LEMMA 1

Proof: Considering the projection of Eq.(13), we can get:

$$\boldsymbol{x}(t+1)=\arg\min_{\boldsymbol{x}\in\Omega} ||\boldsymbol{x}-(\boldsymbol{x}(t)+\beta\cdot\nabla_{\boldsymbol{x}}L_t(\boldsymbol{x}(t),\alpha(t+1)))||_2^2 \,. \tag{48}$$

We expand all the terms, ignore the constant value and divide Eq.(48) by step-size 2β. Thus, we can get the Lemma 1. □

B. PROOF OF LEMMA 2

Proof: We know $\hat{x}^{i,*}$ is the optimal solution for optimization problem P1. Thus, we have:

$$f(\sum_{t=1}^{T}\sum_{i=1}^{M}[\boldsymbol{w}_i^t]^\top\cdot x^{i,*}(t)\cdot\boldsymbol{v}_j)=f(\sum_{t=1}^{T}\sum_{i=1}^{M}[\boldsymbol{w}_i^t]^\top\hat{x}^{i,*}(t)\boldsymbol{v}_j)\,. \tag{49}$$

[41] implies that $[\boldsymbol{w}_i^t]^\top\cdot\boldsymbol{v}_j \le [\widehat{\boldsymbol{w}_i^t}]^\top\cdot\boldsymbol{v}_j$ with prob $(1-\delta)$. Based on the increasing concave function f, we can get Lemma 2 by union bounds [43]. □

C. PROOF OF LEMMA 3

Proof: Eq.(25) is a convex function and the mold is $\frac{1}{2\beta}$, we can get that:

$$\begin{aligned}
&-\boldsymbol{\sigma}_t^T(\boldsymbol{x}(t)-\boldsymbol{x}(t-1))+\frac{||\boldsymbol{x}(t)-\boldsymbol{x}(t-1)||_2^2}{2\beta}\\
&\quad-tf((\sum_{\tau=1}^{t-1}\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^\tau}]^\top x^i(\tau)\boldsymbol{v}_j+\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^t}]^\top x^i(t)\boldsymbol{v}_j)/t)\\
&\le-\boldsymbol{\sigma}_t^T(\boldsymbol{x}^*-\boldsymbol{x}(t-1))+\frac{||\boldsymbol{x}^*-\boldsymbol{x}(t-1)||_2^2}{2\beta}-\frac{||\boldsymbol{x}(t)-\boldsymbol{x}(t-1)||_2^2}{2\beta}\\
&\quad+tf((\sum_{\tau=1}^{t-1}\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^\tau}]^\top x^i(\tau)\boldsymbol{v}_j+\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^t}]^\top x^i(t)\boldsymbol{v}_j)/t)\\
&\le\frac{||\boldsymbol{x}^*-\boldsymbol{x}(t-1)||_2^2}{2\beta}-\frac{||\boldsymbol{x}(t)-\boldsymbol{x}(t-1)||_2^2}{2\beta}\\
&\quad-tf((\sum_{\tau=1}^{t-1}\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^\tau}]^\top x^i(\tau)\boldsymbol{v}_j+\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^t}]^\top x^{i*}(t)\boldsymbol{v}_j)/t)
\end{aligned} \tag{50}$$

Here, $t_1, t_2 \ldots t_n$ are considered as time slots. $\Phi_t = tf((\sum_{\tau=1}^{t-1}\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^\tau}]^\top x^i(\tau)\boldsymbol{v}_j+\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^t}]^\top x^{i*}(t)\boldsymbol{v}_j)/t)$ and $\Psi_t = tf((\sum_{\tau=1}^{t}\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^\tau}]^\top x^i(\tau)\boldsymbol{v}_j)/t)$. Based on Lemma 5 of [23], we have:

$$\sum_{t=1}^{T}(\Phi_t-\Psi_t)\le\frac{\sqrt{T}MD^2}{2}+\frac{3M\sqrt{T}}{2} \tag{51}$$

As f is a concave function, Φ_t and Ψ_{t-1} can be given by:

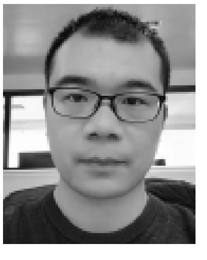
$$\begin{aligned}
\Phi_t &= tf((\sum_{\tau=1}^{t-1}\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^\tau}]^\top x^i(\tau)\boldsymbol{v}_j+\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^t}]^\top x^{i*}(t)\boldsymbol{v}_j)/t)\\
&\ge(t-1)f((\sum_{\tau=1}^{t-1}\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^\tau}]^\top x^i(\tau)\boldsymbol{v}_j)/(t-1))\\
&\quad+f(\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^t}]^\top x^{i*}(t)\boldsymbol{v}_j)\\
&=\Psi_{t-1}+f(\sum_{i=1}^{M}[\widehat{\boldsymbol{w}_i^t}]^\top x^{i*}(t)\boldsymbol{v}_j)
\end{aligned} \tag{52}$$

□

REFERENCES

[1] [Online]. Available: http://www.mturk.com
[2] Y. Tong, L. Chen, Z. Zhou, H. V. Jagadish, L. Shou, and W. Lv, "SLADE: A smart large-scale task decomposer in crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 8, pp. 1588–1601, Aug. 2018.
[3] X. Ren, C.-M. Yu, W. Yu, S. Yang, X. Yang, J. A. McCann, and S. Y. Philip, "LoPub: High-dimensional crowdsourced data publication with local differential privacy," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 9, pp. 2151–2166, Sep. 2018.
[4] Y. Zhang, Y. Gu, M. Pan, N. H. Tran, Z. Dawy, and Z. Han, "Multi-dimensional incentive mechanism in mobile crowdsourcing with moral hazard," *IEEE Trans. Mobile Comput.*, vol. 17, no. 3, pp. 604–616, Mar. 2018.
[5] M. Amich, P. D. Luca, and S. Fiscale, "Accelerated implementation of FQSqueezer novel genomic compression method," in *Proc. 19th Int. Symp. Parallel Distrib. Comput.*, 2020, pp. 158–163.
[6] Y. Tong, J. She, B. Ding, L. Wang, and L. Chen, "Online mobile micro-task allocation in spatial crowdsourcing," in *Proc. ICDE*, May 2016, pp. 49–60.
[7] D. Wedelin, "An algorithm for large scale 0–1 integer programming with application to airline crew scheduling," *Ann. Oper. Res.*, vol. 57, no. 1, pp. 283–301, Dec. 1995.
[8] A. Kittur, J. V. Nickerson, M. Bernstein, E. Gerber, A. Shaw, J. Zimmerman, M. Lease, and J. Horton, "The future of crowd work," in *Proc. IEEE CSCW*, Feb. 2013, pp. 1301–1318.

[9] J. Bragg and A. Kolobov, "Parallel task routing for crowdsourcing," in *Proc. IEEE WWW*, Sep. 2014, pp. 1–11.
[10] D. Deng, C. Shahabi, and U. Demiryurek, "Maximizing the number of worker's self-selected tasks in spatial crowdsourcing," in *Proc. 21st SIGSPATIAL GIS*, Nov. 2013, pp. 314–323.
[11] P. De Luca, A. Galletti, G. Giunta, and L. Marcellino, "Accelerated Gaussian convolution in a data assimilation scenario," in *Proc. Int. Conf. Comput. Sci.*, 2020, pp. 199–211.
[12] S. B. Roy, I. Lykourentzou, S. Thirumuruganathan, S. Amer-Yahia, and G. Das, "Crowds, not drones: Modeling human factors in interactive crowdsourcing," in *Proc. VLDB Workshop*, 2013, pp. 39–42.
[13] A. R. Cardoso and H. Wang, "The online saddle point problem and online convex optimization with knapsacks," *Mach. Learn.*, 2020.
[14] L. Kazemi and C. Shahabi, "GeoCrowd: Enabling query answering with spatial crowdsourcing," in *Proc. 21st SIGSPATIAL GIS*, 2012, pp. 189–198.
[15] E. Ch'ng, S. Cai, T. E. Zhang, and F.-T. Leow, "Crowdsourcing 3D cultural heritage: Best practice for mass photogrammetry," *J. Cultural Heritage Manage. Sustain. Develop.*, vol. 9, no. 1, pp. 24–42, Feb. 2019.
[16] P. Cheng, X. Lian, Z. Chen, R. Fu, L. Chen, and J. Han, "Reliable diversity-based spatial crowdsourcing by moving workers," *Proc. VLDB Endowment*, vol. 8, no. 10, pp. 347–360, 2015.
[17] T. Song, Y. Tong, L. Wang, J. She, B. Yao, L. Chen, and K. Xu, "Trichromatic online matching in real-time spatial crowdsourcing," in *Proc. ICDE*, Apr. 2017, pp. 1009–1020.
[18] Y. Tong, J. She, B. Ding, L. Wang, and L. Chen, "Online mobile microtask allocation in spatial crowdsourcing," in *Proc. IEEE ICDE*, May 2016, pp. 49–60.
[19] G. Goel, A. Nikzad, and A. Singla, "Allocating tasks to workers with matching constraints: Truthful mechanisms for crowdsourcing markets," *J. ACM*, pp. 279–280, 2014.
[20] J. Gao, X. Liu, B. C. Ooi, H. Wang, and G. Chen, "An online cost sensitive decision-making method in crowdsourcing systems," in *Proc. ACM SIGMOD*, 2013, pp. 217–228.
[21] A. Kittur, B. Smus, S. Khamkar, and R. E. Kraut, "Crowdforge: Crowdsourcing complex work," in *Proc. ACM Symp. User Interface Softw. Technol.*, 2011, pp. 43–52.
[22] A. Antos, V. Grover, and C. Szepesvári, "Active learning in multi-armed bandits," in *Algorithmic Learning Theory*. Berlin, Germany: Springer, 2008, pp. 287–302.
[23] H. Xu, Y. Liu, W. C. Lau, T. Zeng, J. Guo, and A. X. Liu, "Online resource allocation with machine variability: A bandit perspective," *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2243–2256, Oct. 2020.
[24] L. T. Thanh, S. Stein, A. Rogers, and N. R. Jennings, "Efficient crowdsourcing of unknown experts using bounded multi-armed bandits," *Artif. Intell.*, vol. 214, pp. 89–111, Sep. 2014.
[25] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, pp. 235–256, May 2002.
[26] H. Kajino, Y. Tsuboi, and H. Kashima, "A convex formulation for learning from crowds," in *Proc. AAAI*, 2012, pp. 1–7.
[27] T. Chen, Q. Ling, and G. B. Giannakis, "An online convex optimization approach to dynamic network resource allocation," 2017, *arXiv:1701.03974*. [Online]. Available: http://arxiv.org/abs/1701.03974
[28] Q. Xu, Q. Huang, and Y. Yao, "Online crowdsourcing subjective image quality assessment," in *Proc. MM*, 2012, pp. 359–368.
[29] P. D. Luca, A. Galletti, G. Giunta, and L. Marcellino, "Recursive filter based GPU algorithms in a data assimilation scenario," *J. Comput. Sci.*, Apr. 2021, Art. no. 101339.
[30] M. Mahdavi, R. Jin, and T. Yang, "Trading regret for efficiency: Online convex optimization with long term constraints," *J. Mach. Learn. Res.*, vol. 13, pp. 2503–2528, Sep. 2012.
[31] L. Zheng and L. Chen, "Mutual benefit aware task assignment in a bipartite labor market," in *Proc. IEEE ICDE*, May 2016, pp. 73–84.
[32] J. Xia, Y. Zhao, G. Liu, J. Xu, M. Zhang, and K. Zheng, "Profit-driven task assignment in spatial crowdsourcing," in *Proc. IJCAI*, Aug. 2019, pp.'1914–1920.
[33] A. Rangi and M. Franceschetti, "Multi-armed bandit algorithms for crowdsourcing systems with online estimation of workers'ability," in *Proc. AAMAS*, 2018, pp. 1345–1352.
[34] A. W. Memon and G. Fursin, "Crowdtuning: Systematizing auto-tuning using predictive modeling and crowdsourcing," in *Proc. PARCO Mini-Symp.*, 2013, pp. 1–13.
[35] S. Agrawal and N. R. Devanur, "Bandits with concave rewards and convex knapsacks," in *Proc. ACM Conf. Econ. Comput.*, Jun. 2014, pp. 989–1006.
[36] P. Auer, "Using confidence bounds for exploitation-exploration tradeoffs," *J. Mach. Learn. Res.*, 2012.
[37] Y. Abbasi-Yadkori, "Improved algorithms for linear stochastic bandits," in *Proc. NIPS*, 2012.
[38] H. Xu, P. Hu, W. C. Lau, Q. Zhang, and Y. Wu, "DPCP: A protocol for optimal pull coordination in decentralized social networks," in *Proc. INFOCOM*, Apr. 2015, pp. 2614–2622.
[39] P. De Luca, A. Galletti, and L. Marcellino, "A Gaussian recursive filter parallel implementation with overlapping," in *Proc. SITIS*, Nov. 2019, pp. 641–648.
[40] C. Jin, P. Netrapalli, and M. I. Jordan, "Accelerated gradient descent escapes saddle points faster than gradient descent," *Proc. Mach. Learn. Res.*, vol. 75, pp. 1042–1085, 2018.
[41] A. Badanidiyuru, R. Kleinberg, and A. Slivkins, "Bandits with knapsacks," *J. ACM*, vol. 65, no. 3, pp. 1–55, Mar. 2018.
[42] P. De Luca and A. Formisano, "Haptic data accelerated prediction via multicore implementation," in *Proc. Sci. Inf. Conf.*, 2020, pp. 110–121.
[43] S. Agrawal and N. Devanur, "Linear contextual bandits with knapsacks," in *Proc. NIPS*, 2016, pp. 3450–3458.
[44] Z. Zheng and N. B. Shroff, "Online multi-resource allocation for deadline sensitive jobs with partial values in the cloud," in *Proc. IEEE INFOCOM*, Apr. 2016, pp. 1–9.
[45] Y. Yu, S. Liu, L. Guo, P. L. Yeoh, B. Vucetic, and Y. Li, "CrowdR-FBC: A distributed fog-blockchains for mobile crowdsourcing reputation management," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8722–8735, Sep. 2020.



QI LI is currently pursuing the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His research interests include job scheduling and resource allocation.

LIJUN CAI received the Ph.D. degree from the College of Computer Science and Electronic Engineering, Hunan University, in 2007. He is currently a Professor with Hunan University. His research interests include bioinformatics, cloud computing, and big data scheduling and management.