$Crowd^2$: Multi-agent Bandit-based Dispatch for Video Analytics upon Crowdsourcing

Yu Chen[†], Sheng Zhang[*†], Yuting Yan[†], Yibo Jin[†], Ning Chen[†], Mingtao Ji[†], and Mingjun Xiao[§]
[†]State Key Lab. for Novel Software Technology, Nanjing University, P.R. China
[§]School of Computer Science and Technology, University of Science and Technology of China, P.R. China
Email: yu.chen@smail.nju.edu.cn, sheng@nju.edu.cn, {yuting.yan, yibo.jin, ningc, jmt}@smail.nju.edu.cn, xiaomj@ustc.edu.cn

***Abstract*—Many crowdsourcing platforms are emerging, leveraging the resources of recruited workers to execute various outsourcing tasks, mainly for those computing-intensive video analytics with high quality requirements. Although the profit of each platform is strongly related to the quality of analytics feedback, due to the uncertainty on diverse performance of workers and the conflicts of interest over platforms, it is non-trivial to determine the dispatch of tasks with maximum benefits. In this paper, we design a decentralized mechanism for a Crowd of Crowdsourcing platforms, denoted as $Crowd^2$, optimizing the worker selection to maximize the social welfare of these platforms in a long-term scope, under the consideration of both proportional fairness and dynamic flexibility. Concretely, we propose a video analytics dispatch algorithm based on multi-agent bandit, for which the more accurate profit estimates are attained via the decoupling of multi-knapsack based mapping problem. Via rigorous proofs, a sub-linear regret bound for social welfare of crowdsourcing profits is achieved while both fairness and flexibility are ensured. Extensive trace-driven experiments demonstrate that $Crowd^2$ improves the social welfare by 36.8%, compared with other alternatives.**

I. Introduction

In the past decade, with the development of mobile crowdsourcing [1–4], more and more crowdsourcing platforms (e.g., Amazon Mechanical Turk [5], CrowdFlower [6]) are emerging, and they leverage the resource of crowdsourcing workers, recruited from mobile users, to execute various crowdsourcing tasks [7–9], such as image labelling, mobile sensing and traffic prediction. Among those tasks, video analytics [10–13] including object detection, identification and tracking, is attracting much attention, with the reward highly depending on worker's performance like result quality [14] and execution cost [15]. In result, the platforms need to select suitable workers and then dispatch video analytics tasks to them for maximum profits.

However, as shown in Fig. 1, it is non-trivial to optimally dispatch the tasks for multiple competing platforms and various workers without priori performance profiles in video analytics. Specifically, dispatch for video analytics upon multiple crowdsourcing platforms faces these crucial challenges:

First of all, workers' performance in terms of video analytics quality and energy consumption is uncertain and time-varying. As demonstrated in our case studies later, although the relationship between video analytics accuracy and configuration (i.e., frame rate and resolution) can be modeled [12, 16], the analytics accuracy varies with video content even when the video analytics configuration is fixed. Besides, we observe that the energy consumption for video analytics on devices fluctuates over time as well [17, 18]. The stochastic fluctuations in accuracy and energy consumption hamper the platforms from estimating the workers' performance accurately. Furthermore, it is difficult to achieve a trade-off between analytics result quality and execution cost for maximum profits.

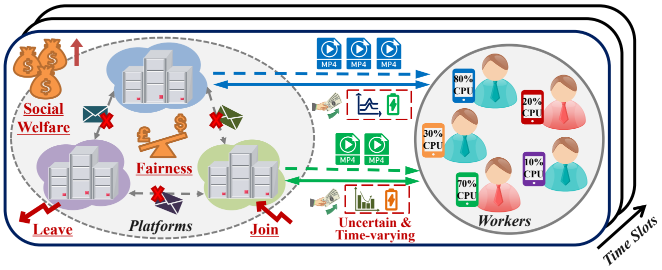
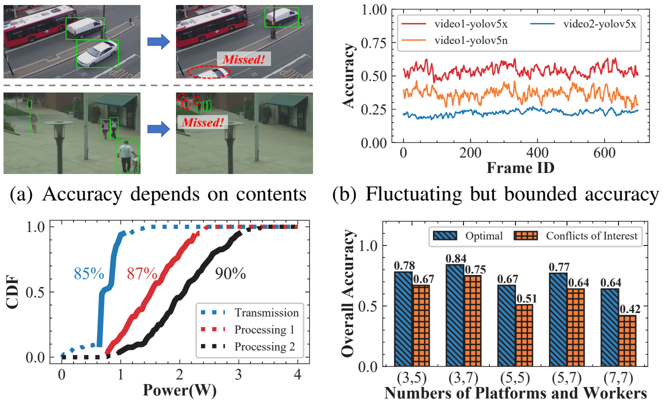


Fig. 1. Dispatch for video analytics upon crowdsourcing platforms

Second, maximizing the social welfare (i.e., the total profits) should also consider the conflicts of interest among platforms. For platforms with similar video analytics outsourcing tasks, they select appropriate workers from a shared crowdsourcing worker set [19]. Each worker can be recruited by multiple platforms and execute multiple tasks in each time period [20, 21]. However, due to the restricted capability (e.g., CPU, RAM) of devices [22], the workers inevitably abandon some task requests if overloaded [23]. Further, since each platform is considered rational and selfish [24], applying a centralized mechanism [25] to guide the exchange of private information among multiple platforms is not practical. Therefore, a well-designed decentralized mechanism is desired to coordinate the worker recruitment for video analytics across platforms.

Third, fairness and flexibility should be taken into account for platforms while designing the mechanism. When maximizing the social welfare, the platforms observing higher profits tend to assign more tasks to workers with higher analytics accuracy and lower energy consumption, thus blocking other platforms from better workers [26]. As studied in prior works [27, 28], the lack of consideration for fairness aggravates the profit differences among platforms. Besides, the platforms can dynamically join or leave at any time [29], leading to a varying number of platforms. The joining platforms require extra time to learn about uncertain workers, and the leaving of platforms affects the overall learning progress, which both pose a challenge to video analytics task dispatch.

[*]The corresponding author is Sheng Zhang (sheng@nju.edu.cn). This work was supported in part by NSFC (61872175, 61832008), the Fundamental Research Funds for the Central Universities (2022300297), and Collaborative Innovation Center of Novel Software Technology and Industrialization. We also thank Zili Meng for his valuable suggestions for this work.



(a) Accuracy depends on contents (b) Fluctuating but bounded accuracy

(c) CDF of consumed energy (d) Social welfare for crowdsourcing

Fig. 2. Motivation for mechanism $Crowd^2$

Existing works fall insufficient for tackling the above challenges. Some works [30–33] focus on determining worker selection for maximum crowdsourcing utility in various scenarios, but few of them optimize the social welfare for platforms that collaboratively recruit workers in a decentralized way. Other works [10–13] research the configuration adaptation for video analytics, but ignore the impacts of video analytics on energy consumption and crowdsourcing profits. The rest [34–37] investigate the stochastic changes in analytics quality or energy consumption, but do not consider the shared resources for multiple agents while involving the fairness and flexibility.

In this paper, we propose a decentralized mechanism for a Crowd of Crowdsourcing platforms[1] dispatching video analytics tasks to workers, named as $Crowd^2$, which overcomes the previous challenges. Under mechanism $Crowd^2$, we first formulate an online optimization problem with the objective of maximizing the social welfare for multiple platforms in a long-term scope. Specifically, the crowdsourcing profit depends on video analytics accuracy and energy consumption, which are time-varying and uncertain. Besides, due to the limited computing capacity of workers' devices, the computation demand on each worker from multiple platforms is also constrained.

We then design a multi-agent bandit-based decentralized online algorithm for video analytics dispatch upon multiple platforms, capturing the stochastic changes on the crowdsourcing profit. In order to attain more accurate profit estimates for the bandit, we model the original mapping problem in bandit as a multi-knapsack problem, which is decoupled into a series of knapsack sub-problems for each worker. Furthermore, we guarantee the fairness among platforms by means of proportional fairness maximization and investigate the flexibility for $Crowd^2$ in terms of the varying platform number. Via rigorous proofs, by employing our proposed algorithm based on bandit for maximum social welfare of video analytics crowdsourcing, which takes the fairness and flexibility into consideration, the proportional fairness among platforms is ensured, and a sub-linear regret bound for profits is achieved. Concretely, a regret of $\mathcal{O}(\log T)$ is upper bounded by our proposed algorithm, and when further considering the flexibility for $Crowd^2$, an $\mathcal{O}(T^{\xi})$ regret bound is accomplished, where $0 < \xi < 1$.

TABLE I
COMPARISON BETWEEN RELATED WORKS ON CROWDSOURCING

Existing Works	Social Welfare	Fairness	Flexibility	Uncertainty
DRL-MTVCS[20]	✓	✓	×	✓
MCE2C[21]	✓	×	✓	✓
Centurion[24]	✓	×	×	×
LOL[32]	×	×	✓	✓
ON-DYN[33]	×	×	✓	×
EUWR[35]	×	×	×	✓
$Crowd^2$	✓	✓	✓	✓

Extensive trace-driven experiments using the videos derived from PANDA dataset [38] testify to the superiority of $Crowd^2$ in social welfare for YOLOv5-based [39] video analytics crowd-sourcing compared with other methods. Specifically, the social welfare for platforms averagely increases 36.8% with 25.6% reduction in the fairness gap.

II. SYSTEM MODEL

A. Motivating Case Studies for Video Crowdsourcing

Learnability for Video Analytics. As investigated in some existing works [12, 16], the relationship between video analytics accuracy and configuration (i.e., frame rate, resolution) can be modeled as concave functions. However, case studies with CentOS 7.6, Python 3.7 and YOLOv5 models [39] show that even when the configuration is fixed, the accuracy varies depending on the video content, as shown in Fig. 2(a). Besides, the energy consumption for downloading and executing the video analytics tasks also fluctuates over time [17, 18]. Fortunately, however, we notice that the changes in analytics accuracy and energy consumption are bounded up and down within a range, as demonstrated in Figs. 2(b) and 2(c). Therefore, their expected values can reflect workers' video analytics performance, motivating us to design an algorithm upon exploration-and-exploitation to determine the worker selection.

Social Welfare of Crowdsourcing. For some specific video analytics tasks, only a limited number of workers are available for recruitment [19], and the computing capacity of their devices is also restricted [22, 23] within a specified time period. Thus, competition for better workers incurs conflicts of interest among platforms and then reduces the overall video analytics quality, thereby resulting in lower social welfare. As illustrated in Fig. 2(d), case studies based on PANDA Dataset [38] present the reduction in overall video analytics accuracy caused by platforms' intensely competing for workers. On the other hand, the lack of consideration for fairness (controlled competition), flexibility (scheduling for platforms in and out) and uncertainty (treating stochastic inputs), as in Table I, can have bad effects on social welfare in the long run [28]. Thus, it is of great necessity to tackle platforms' conflicts of interest for maximum social welfare, with uncertainty, while ensuring fairness and flexibility under mechanism $Crowd^2$.

B. System Settings and Models

We summarize the important notations in Table II.

Video Crowdsourcing. Consider the mechanism $Crowd^2$, with the overall time horizon divided into many time slots, represented as $\boldsymbol{\mathcal{T}} = \{1, ..., T\}$. We denote the crowdsourcing platform set as $\boldsymbol{\mathcal{N}} = \{1, ..., N\}$, where the platforms receive

[1] "A Crowd of Crowdsourcing platforms" means multiple platforms.

TABLE II
MAJOR NOTATIONS USED FOR MODEL

Input	Description
$\boldsymbol{\mathcal{N}}$	$\{1, 2, ..., N\}$, set of crowdsourcing platforms
$\boldsymbol{\mathcal{M}}$	$\{1, 2, ..., M\}$, set of crowdsourcing workers
$\boldsymbol{\mathcal{T}}$	$\{1, 2, ..., T\}$, set of time slots for the overall time horizon
$\boldsymbol{\mathcal{N}}_{m,t}$	Set of platforms dispatching analytics tasks to worker m in time slot t
c_n	Computation demand of platform n in each time slot
C_m	Computing capacity for worker m in each time slot
$a_{n,m}$	Video analytics accuracy for platform n recruiting worker m
$e_{n,m}$	Energy consumption for worker m recruited by platform n
$\tilde{r}_{n,m}$	Observed reward of platform n dispatching analytics task to worker m
Decision	**Description**
$x_{n,t}$	Worker selection for platform $n \in \boldsymbol{\mathcal{N}}$ in time slot $t \in \boldsymbol{\mathcal{T}}$

the video analytics tasks from the requestors and then dispatch them to the recruited workers. The set of available workers[2] are denoted as $\boldsymbol{\mathcal{M}} = \{1, ..., M\}$, and the platforms need to determine the worker selection for each time slot. We use $x_{n,t}$ to represent the worker selected by platform n in time slot t, and $\boldsymbol{x}_t = (x_{1,t}, ..., x_{N,t})$ to represent the overall decision for all platforms. Besides, we let $\boldsymbol{\mathcal{N}}_{m,t}$ be the set of platforms that dispatch the tasks to the worker m in time slot t.

Analytics Accuracy. According to the existing works [12, 13, 32], we attain two important observations for profiling the relationship between video analytics configuration and accuracy: a) frame rate sampling and frame resolution adaptation impact accuracy independently; b) relationship between resolution/frame rate and accuracy can be modeled as a concave function. Based on them, the accuracy for video analytics dispatched to worker m by platform n can be calculated as

$$a_{n,m} = \phi_m(f_n)\epsilon_m(s_n), \tag{1}$$

where the concave functions $\phi_m(f)$ and $\epsilon_m(s)$ mean the accuracies with respect to frame rate f and resolution s when the video analytics task is executed by worker m. Besides, the frame rate and resolution of the video from platform n are represented as f_n and s_n, respectively.

Energy Consumption. For crowdsourcing workers with mobile devices, the battery is one of the most concerns [32] because it is not convenient to recharge the devices. Since the energy consumption mainly consists of transmission energy and processing energy, we calculate the energy consumption for worker m when executing the task from platform n as

$$e_{n,m} = (\gamma_m + \mu_m) f_n \alpha(s_n)^2, \tag{2}$$

where $\alpha(s)^2$ is used to represent the data size of a frame with resolution s, and α is a constant [12, 40]. Besides, we let γ_m and μ_m be the energy consumption for downloading and processing, respectively, one bit of video data for analytics [41].

Profit of Crowdsourcing. For each platform, we use $V(n)$ to denote the intrinsic task value (e.g., the paid money from the task requestor) in each time slot. Additionally, the quality of analysis results and energy consumption are the other two important factors, which influence crowdsourcing profit. Thus, we calculate the crowdsourcing profit for each platform n which dispatches the video analytics task to worker m as

$$r_{n,m} = V(n) + A(n,m) - E(n,m), \tag{3}$$

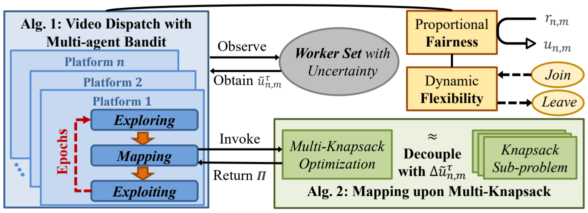

Alg. 1: Video Dispatch with Multi-agent Bandit
Platform n
Platform 2
Platform 1
Epochs
Exploring
Mapping
Exploiting
Observe
Obtain $\tilde{u}^{\tau}_{n,m}$
Worker Set with Uncertainty
Invoke
Return Π
Proportional Fairness
$r_{n,m}$
$u_{n,m}$
Dynamic Flexibility
Join
Leave
Multi-Knapsack Optimization
≈
Decouple with $\Delta\tilde{u}^{\tau}_{n,m}$
Knapsack Sub-problem
Alg. 2: Mapping upon Multi-Knapsack

Fig. 3. Design of our proposed decentralized online algorithm

where we let $A(n,m) = G_n(a_{n,m})$ represent the potential benefits from the video analytics quality (e.g., higher accuracy will attract more video analytics task requests), and a concave function $G_n(a)$ is used as the revenue function with respect to accuracy a for crowdsourcing platform n [42]. Besides, we let $E(n,m) = \omega_m e_{n,m}$ represent platform n's payment to worker m, which mainly covers the execution cost for energy consumption, and ω_m denotes the payment priced by worker m for consuming each unit of energy [43].

Restricted Ability of Workers. Similar to prior works [22, 23], there exists a computing capacity constraint C_m for each worker m in each time slot t. When the computation demand on worker m exceeds its computing capacity, some of the tasks dispatched from platforms will be abandoned, thereby resulting in $r_{n,m} = 0$. Thus, for each worker m, it holds that

$$\sum\nolimits_{n \in \boldsymbol{\mathcal{N}}_{m,t}} c_n \leq C_m, \tag{4}$$

where we assume that the computation demand is proportional to the input video data size, as considered in [44], and it is denoted as $c_n = \beta_n f_n \alpha\left(s_n\right)^2$ using proportionality factor β_n.

C. Problem Formulation

Due to the uncertainty and various types of accuracy model functions (i.e., ϵ_m and ϕ_m) during video analytics and energy consumption rates (i.e., γ_m and μ_m) for crowd workers, each platform can only observe an independent identically distributed (i.i.d.) random reward $\tilde{r}_{n,m}(t)$ in each time slot t, with $r_{n,m} = \mathbb{E}\left[\tilde{r}_{n,m}(t)\right], \tilde{r}_{n,m}(t) \in \left[r^{\min}, r^{\max}\right], \forall n \in \boldsymbol{\mathcal{N}}, \forall m \in \boldsymbol{\mathcal{M}}$, where the positive value $r^{\min}$ and $r^{\max}$ represent the lower and upper reward bounds, respectively. To maximize the social welfare for all platforms in $Crowd^2$, we formulate the utilitarian compound reward optimization problem as

$$\begin{aligned} \mathbb{P}: \max_{\{\boldsymbol{x}_t | t \in \boldsymbol{\mathcal{T}}\}} & \sum\nolimits_{t \in \boldsymbol{\mathcal{T}}} \sum\nolimits_{n \in \boldsymbol{\mathcal{N}}} \mathbb{E}[\tilde{r}_{n,x_{n,t}}(t)] \\ \text{s.t.} & \sum\nolimits_{n \in \boldsymbol{\mathcal{N}}_{m,t}} c_n \leq C_m, \forall m \in \boldsymbol{\mathcal{M}}, \forall t \in \boldsymbol{\mathcal{T}}. \end{aligned}$$

Problem Challenge. The major difficulty in obtaining the optimal solution for $\mathbb{P}$ is the uncertainty of the varying observed reward $\tilde{r}_{n,x_{n,t}}(t)$. Thus, an online approach is required to efficiently determine the worker selection for platforms by exploring and exploiting the observed rewards. Besides, the fairness and flexibility for mechanism $Crowd^2$ are also of great importance, and we are expected to take them into consideration when designing the algorithm.

III. DECENTRALIZED ONLINE ALGORITHM DESIGN

The design of our proposed algorithm for $Crowd^2$ is shown in Fig. 3. To capture the stochastic changes on the reward,

[2] $\boldsymbol{\mathcal{M}}$ denotes the set of all workers. However, in the real world, each platform may only reach a subset of workers, and we will explain it in Section IV.

we first present the video analytics dispatch based on multi-agent bandit in **Alg. 1**, considering the fairness and flexibility. Further, to obtain more accurate reward estimates for bandit, **Alg. 2** is invoked to decouple the original mapping problem into a series of knapsack sub-problems for each worker.

A. Problem Transformation for Fairness

In order to maximize the utilitarian compound reward in $\mathbb{P}$, the platforms observing higher rewards tend to dispatch more tasks to the workers with higher video analytics quality and lower energy consumption, thereby blocking other platforms from the "better" workers and then causing the unfair worker recruitment for video analytics in $Crowd^2$. To avoid the above undesirable outcome, we use the proportional fairness maximization [27, 45] to guarantee fairness among platforms. We first present the definition of proportional fairness as

Definition 1 (Proportional Fairness)**.** In time slot t, a decision $\boldsymbol{x}_t^* = (x_{1,t}^*, ..., x_{N,t}^*)$ has the property of proportional fairness if and only if for any other feasible decision $\boldsymbol{x}_t' = (x_{1,t}', ..., x_{N,t}')$, the following inequation always holds:

$$\sum\nolimits_{n\in\boldsymbol{\mathcal{N}}}(r_{n,x_{n,t}'} - r_{n,x_{n,t}^*})/(r_{n,x_{n,t}^*}) \leq 0. \tag{5}$$

The intuition behind proportional fairness is as follows: for the platform n selecting the worker of the highest reward under the "fairest" decision $\boldsymbol{x}_t^*$, if we try to obtain a more "fair" decision that reduces platform n's reward and improves others', we will find that no more "fair" decision is feasible because the highest reward $r_{n,x_{n,t}^*}$ serves as the denominator in inequation (5), and the decrease in it will violate the inequation. Similarly, the above intuition can also be used as the explanation for the platform of the lowest reward. Therefore, the decision $\boldsymbol{x}_t^*$ based on the definition of proportional fairness is indeed a "fair" decision. To maximize social welfare while ensuring the fairness, we design the utility function as

$$u_{n,x_{n,t}}(r_{n,x_{n,t}}) = \ln(1 + \rho r_{n,x_{n,t}}), \rho > 0, \tag{6}$$

for each platform according to **Proposition 1** [29] as follows.

Proposition 1. *Set* $u_{n,x_{n,t}}(r_{n,x_{n,t}}) = \ln(1+\rho r_{n,x_{n,t}}), \rho > 0,$ *and then the optimal solution* $\boldsymbol{x}_t^*$ *for* $\max_{\boldsymbol{x}_t} \sum_{n\in\boldsymbol{\mathcal{N}}} u_{n,x_{n,t}}$ *satisfies the inequation (5) of Proportional Fairness.*

Proof. See Appendix A, upon the concavity of Eq. (6). □

Problem Transformation. Based on the above proposition, we transform problem $\mathbb{P}$ into $\mathbb{P}_1$ and modify its optimization objective, which takes the proportional fairness into consideration. Thus, the problem $\mathbb{P}_1$ is formulated as

$$\begin{aligned} \mathbb{P}_1 : \max_{\{\boldsymbol{x}_t | t\in\boldsymbol{\mathcal{T}}\}} & \sum\nolimits_{t\in\boldsymbol{\mathcal{T}}} \sum\nolimits_{n\in\boldsymbol{\mathcal{N}}} \mathbb{E}[\tilde{u}_{n,x_{n,t}}(t)] \\ \text{s.t. } & \tilde{u}_{n,x_{n,t}}(t) = \ln(1 + \rho\tilde{r}_{n,x_{n,t}}(t)), \\ & \sum\nolimits_{n\in\boldsymbol{\mathcal{N}}_{m,t}} c_n \leq C_m, \forall m \in \boldsymbol{\mathcal{M}}, \forall t \in \boldsymbol{\mathcal{T}}. \end{aligned}$$

Accordingly, instead of $\tilde{r}_{n,m}(t)$, we observe the i.i.d. random reward value $\tilde{u}_{n,m}(t)$, with $u_{n,m} = \mathbb{E}\left[\tilde{u}_{n,m}(t)\right], \tilde{u}_{n,m}(t) \in \left[u^{\min}, u^{\max}\right], \forall n \in \boldsymbol{\mathcal{N}}, \forall m \in \boldsymbol{\mathcal{M}}$, where the positive $u^{\min}$ and $u^{\max}$ are the lower and upper reward bounds, respectively. Based on the proportional fair reward $u_{n,x_{n,t}}$, which contributes to the fairness for $Crowd^2$, we propose the video analytics dispatch algorithm in the following subsections.

Algorithm 1: Video Dispatch with Multi-agent Bandit

```
   Input: $T_{\text{explore}}=\lceil\frac{25(N_{\max})^2(u^{\max}-u^{\min})^2 M}{2(\delta_{\min})^2}\rceil, T_{\text{exploit}}=2^\tau$
1  $\tilde{u}_{n,m}^\tau \leftarrow 0, \forall n \in \boldsymbol{\mathcal{N}}, \forall m \in \boldsymbol{\mathcal{M}}, \forall\tau = 0, 1, 2, ...;$
2  for epoch $\tau = 1$ to $\tau_T$ do
       // Exploring
3      for time slot $t = 1$ to $T_{explore}$ do
4          for batch $b = 1$ to $\lceil\frac{N}{N_{min} M}\rceil$ do
5              Each platform $n$ in batch $b$ sends video analytics
                 task to worker $\lfloor((n+t)\%(N_{\min}M))/N_{\min}\rfloor+1;$
6              $\tilde{u}_{n,m}^\tau \leftarrow (\tilde{u}_{n,m}^{\tau-1} \times (\tau - 1) + \tilde{u}_{n,m})/\tau;$
       // Mapping
7      Call Alg. 2 with input $\{\tilde{u}_{n,m}^\tau\}$ and $\boldsymbol{\Pi} = \boldsymbol{0}$;
       // Exploiting
8      for time slot $t = 1$ to $T_{exploit}$ do
9          Platforms dispatch video analytics tasks upon $\boldsymbol{\Pi}$;
```

B. Video Analytics Dispatch with Multi-agent Bandit

The uncertainty and variability of rewards motivate us to design an exploration-and-exploitation based algorithm. As some existing prior studies [34–37], multi-armed bandit methods, which involve the tradeoff between exploration and exploitation, are usually leveraged to learn the random rewards from multiple "bandits". However, since the platforms recruit workers in a decentralized way, directly applying the multi-armed bandit method to each platform may lead to conflicts of interest. Furthermore, the intrinsic task values of all platforms are different, which requires an extended multi-armed bandit method considering the heterogeneity of platforms. Therefore, we propose a multi-agent multi-armed bandit based decentralized algorithm to tackle the video analytics dispatch for crowdsourcing platforms in $Crowd^2$ as shown in **Alg. 1**.

Regret. When using the bandit-based **Alg. 1** for $\mathbb{P}_1$, the *regret* to quantify the dispatch performance is defined as

$$\mathcal{R}(T) = T\sum\nolimits_{n\in\boldsymbol{\mathcal{N}}} u_{n,x_{n,t}^*} - \sum\nolimits_{t=1}^{T} \sum\nolimits_{n\in\boldsymbol{\mathcal{N}}} \mathbb{E}[\tilde{u}_{n,x_{n,t}}(t)], \tag{7}$$

where the decision $\{x_{n,t}^*, n \in \boldsymbol{\mathcal{N}}, t \in \boldsymbol{\mathcal{T}}\}$ is optimal for $\mathbb{P}_1$, and $\{x_{n,t}, n \in \boldsymbol{\mathcal{N}}, t \in \boldsymbol{\mathcal{T}}\}$ is obtained from **Alg. 1**. We then divide the overall time horizon $\boldsymbol{\mathcal{T}}$ into multiple epochs $\{1, \ldots, \tau_T\}$, where each epoch consists of a variable number of time slots and τ_T represents the last epoch index. In order to achieve the crucial balance between video analytics dispatch exploration and exploitation, we construct three phases for each epoch, including exploring, mapping, and exploiting as follows.

Exploring. Exploration phase occupies T_{explore} time slots in each epoch, and we propose a batch-based dispatch scheme for reward exploration. In each batch, a group of $N_{\min}M$ platforms dispatch their tasks to the corresponding workers as

$$x_{n,t} \leftarrow \lfloor((n + t)\%(N_{\min}M))/N_{\min}\rfloor + 1, \tag{8}$$

which is illustrated in line 5. For the group size $N_{\min}M$, we set $N_{\min} = \frac{\min_{m\in\boldsymbol{\mathcal{M}}}\{C_m\}}{\max_{n\in\boldsymbol{\mathcal{N}}}\{c_n\}}$ as the ratio between the minimum worker capacity and maximum platform demand such that any worker computing capacity will not be violated in exploration. The estimated reward $\tilde{u}_{n,m}^\tau$ is initialized as 0 and updated as

$$\tilde{u}_{n,m}^\tau \leftarrow (\tilde{u}_{n,m}^{\tau-1} \times (\tau - 1) + \tilde{u}_{n,m})/\tau, \tag{9}$$

Algorithm 2: Mapping upon Multi-Knapsack Problem

```
   Input: $\{\tilde{u}^{\tau}_{n,m}\}, \mathbf{\Pi} = 0$
1  for worker $m = 1$ to $M$ do
2      for platform $n = 1$ to $N$ do
3          if $\Pi_n \neq 0$ then
4              $\Delta\tilde{u}^{\tau}_{n,m} \leftarrow \tilde{u}^{\tau}_{n,m} - \tilde{u}^{\tau}_{n,\Pi_n}$;
5          else
6              $\Delta\tilde{u}^{\tau}_{n,m} \leftarrow \tilde{u}^{\tau}_{n,m}$;
7      Worker $m$ tackles $\mathbb{P}_3$ with input $\{\Delta\tilde{u}^{\tau}_{n,m}\}$;
8      for platform $n = 1$ to $N$ do
9          if worker $m$ accepts platform $n$'s task then
10             Update $\Pi_n \leftarrow m$;
   Output: $\mathbf{\Pi}$
```

by calculating the expectation of all explored observations from the beginning to the current epoch τ in line 6.

Mapping and Exploiting. After exploring the estimated rewards, we invoke **Alg. 2** with input[3] $\{\tilde{u}^{\tau}_{n,m}\}$ to yield the mapping result $\mathbf{\Pi} = \{\Pi_n\}$, which will be elaborated in the next subsection. At last, exploitation phase in each epoch lasts for $T_{\text{exploit}} = 2^{\tau}$ time slots, which can also be replaced with other exponential forms (explained in Appendix E), and all platforms dispatch video analytics tasks to the workers obeying the mapping result $\mathbf{\Pi}$ from **Alg. 2** to fully exploit rewards.

C. Mapping upon Multi-Knapsack Problem

For mapping phase, the goal is to obtain the video analytics dispatch mapping $\mathbf{\Pi}$ from the platforms to workers based on estimated rewards $\{\tilde{u}^{\tau}_{n,m}\}$, and it can be modeled as a multi-knapsack problem for each time slot t [46]:

$$\begin{aligned} \mathbb{P}_2: \ \max_{\boldsymbol{x}_t} \ & \sum\nolimits_{n\in\boldsymbol{\mathcal{N}}} \tilde{u}^{\tau}_{n,x_{n,t}} \\ \text{s.t.} \ & \sum\nolimits_{n\in\boldsymbol{\mathcal{N}}_{m,t}} c_n \leq C_m, \forall m \in \boldsymbol{\mathcal{M}}, \end{aligned}$$

which is usually considered NP-hard, and cannot be optimally solved in polynomial time [47]. In order to tackle the above intractable challenge, we design a multi-knapsack based mapping algorithm as shown in **Alg. 2**.

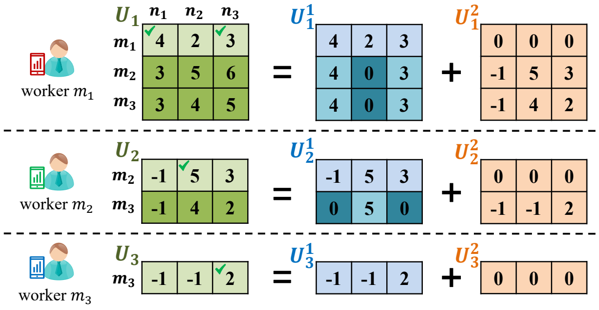
Decoupling Multi-knapsack. We first decouple the multi-knapsack problem $\mathbb{P}_2$ into a series of knapsack sub-problems based on the reward improvement $\Delta\tilde{u}^{\tau}_{n,m}$ defined as

$$\Delta\tilde{u}^{\tau}_{n,m} \leftarrow \begin{cases} \tilde{u}^{\tau}_{n,m} - \tilde{u}^{\tau}_{n,\Pi_n}, & \text{if } \Pi_n \neq 0, \\ \tilde{u}^{\tau}_{n,m}, & \text{otherwise}, \end{cases} \tag{10}$$

which intuitively marks the improved reward if platform n's video analytics task is dispatched to another worker. We then represent each knapsack sub-problem with the optimization objective using $\Delta\tilde{u}^{\tau}_{n,m}$ for worker m as

$$\begin{aligned} \mathbb{P}_3: \ \max_{\boldsymbol{x}_t} \ & \sum\nolimits_{n\in\boldsymbol{\mathcal{N}}_{m,t}} \Delta\tilde{u}^{\tau}_{n,m} \\ \text{s.t.} \ & \sum\nolimits_{n\in\boldsymbol{\mathcal{N}}_{m,t}} c_n \leq C_m. \end{aligned}$$

Iteratively Updating $\mathbf{\Pi}$. For each worker m, sub-problem $\mathbb{P}_3$ can be efficiently solved using branch-and-bound method [48, 49] with $\{\Delta\tilde{u}^{\tau}_{n,m}\}$ as shown in line 7, i.e., gradually improving the video analytics task dispatch reward. If the solution to $\mathbb{P}_3$ results in platform n dispatching its task to worker m, we update $\Pi_n \leftarrow m$ in line 10 and then recalculate the values for $\{\Delta\tilde{u}^{\tau}_{n,m}\}$ when tackling the next knapsack sub-problem for worker $m \leftarrow m+1$. Finally, the output mapping $\mathbf{\Pi}$ is used as a basis for the exploiting phase in **Alg. 1**. Notably, the mapping phase has a length of M time slots, which is corresponding to the M crowdsourcing workers.



$$\begin{array}{c|ccc} U_1 & n_1 & n_2 & n_3 \\ \hline m_1 & 4\checkmark & 2 & 3\checkmark \\ m_2 & 3 & 5 & 6 \\ m_3 & 3 & 4 & 5 \end{array} = \begin{array}{ccc} \multicolumn{3}{c}{U_1^1} \\ 4 & 2 & 3 \\ 4 & 0 & 3 \\ 4 & 0 & 3 \end{array} + \begin{array}{ccc} \multicolumn{3}{c}{U_1^2} \\ 0 & 0 & 0 \\ -1 & 5 & 3 \\ -1 & 4 & 2 \end{array}$$

$$\begin{array}{c|ccc} U_2 & & & \\ \hline m_2 & -1 & 5\checkmark & 3 \\ m_3 & -1 & 4 & 2 \end{array} = \begin{array}{ccc} \multicolumn{3}{c}{U_2^1} \\ -1 & 5 & 3 \\ 0 & 5 & 0 \end{array} + \begin{array}{ccc} \multicolumn{3}{c}{U_2^2} \\ 0 & 0 & 0 \\ -1 & -1 & 2 \end{array}$$

$$\begin{array}{c|ccc} U_3 & & & \\ \hline m_3 & -1 & -1 & 2\checkmark \end{array} = \begin{array}{ccc} \multicolumn{3}{c}{U_3^1} \\ -1 & -1 & 2 \end{array} + \begin{array}{ccc} \multicolumn{3}{c}{U_3^2} \\ 0 & 0 & 0 \end{array}$$

Fig. 4. Running example for **Alg. 2** with 3 platforms and 3 workers

Running Example. We give an example for **Alg. 2** in Fig. 4, where there are 3 platforms and 3 workers. Assume that the task computation demand of the 3 platforms is $(2, 3, 1)$, and the computing capacity of the 3 workers is $(4, 3, 3)$. We show $\{\Delta\tilde{u}^{\tau}_{n,m}\}$ in matrix U_1, and $\tilde{u}^{\tau}_{n,m}$ is located in the m-th row and the n-th column of U_1. For worker m_1, we attain the optimal solution for $\mathbb{P}_3$ as $S_{m_1} = \{n_1, n_3\}$, which means that worker m_1 accepts the tasks from platform n_1 and n_3. After that, to efficiently update the improved reward matrix as lines 3-6 in **Alg. 2**, we build the matrix U_1^1 composed of 3 parts. The first part is the m_1-th row, where the elements are the same as U_1. In the rest of U_1^1, the second part is the columns n_1 and n_3, where the elements are the same as row m_1. The third part is the remaining part, which is filled with zeros. Based on U_1^1, we can easily calculate the matrix $U_1^2 = U_1 - U_1^1$, and then the updated improved reward matrix U_2 can be quickly obtained by removing the row m_1 in U_1^2. Similarly, we can attain the optimal solutions of $\mathbb{P}_3$ for worker m_2 and m_3. Thus, the final output $\mathbf{\Pi}$ of **Alg. 2** is $\{\Pi_1 = n_1, \Pi_2 = n_2, \Pi_3 = n_3\}$.

D. Ensured Flexibility Produced by $Crowd^2$

Furthermore, in our proposed mechanism $Crowd^2$, the platforms may dynamically join or leave the platform set, which will lead to a varying platform number N and pose a challenge to our design. Thus, it is essential to consider the flexibility of $Crowd^2$, which allows the entry and exit of platforms.

(i) We first discuss the impact of platforms' leaving. For the case of leaving, since the remaining platforms have explored sufficient samples of rewards, which are utilized for the later phases of mapping and exploiting, our proposed multi-agent multi-armed bandit-based video analytics dispatch algorithm still works and achieves a logarithmic regret bound with respect to T, which is rigorously proved in Section IV.

(ii) For case of joining, we denote the platform number in epoch τ as N_τ. Different from the case of leaving, platforms' dynamic entering causes an unbounded exploration error prob-

[3] For ease of expression, let $\{\tilde{u}^{\tau}_{n,m}\}$ and $\{\Delta\tilde{u}^{\tau}_{n,m}\}$ represent the sets of $\tilde{u}^{\tau}_{n,m}$ and $\Delta\tilde{u}^{\tau}_{n,m}$ for $\{\forall n \in \boldsymbol{\mathcal{N}}, m \in \boldsymbol{\mathcal{M}}, \tau \in \{1, ..., \tau_T\}\}$, respectively.

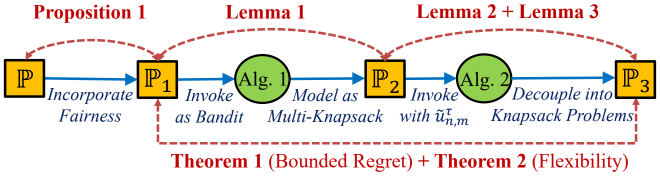


Fig. 5. Roadmap for theoretical analysis

ability because the newly joining platforms have not explored enough times to obtain accurate estimates for proportional fair rewards, which will decrease the performance of mapping and exploiting. However, as shown in **Theorem 2** of the next section, **Alg. 1** with $N = N_{\tau}$ still ensures the flexibility for $Crowd^2$ and achieves a sub-linear regret with respect to T.

IV. Performance Analysis

The roadmap for theoretical analysis is illustrated in Fig. 5. The main result about the regret of social welfare in $Crowd^2$ is shown in **Theorem 1**. Furthermore, the regret of social welfare considering flexibility is shown in **Theorem 2**.

Lemma 1 (Bounded Exploration Error)**.** *After the τ-th epoch of exploration in **Alg. 1**, error probability is bounded:*

$$\Pr(|\tilde{u}^{\tau}_{n,m} - u_{n,m}| > \delta_{\min}/(5N_{\max})) \le 2NMe^{-\tau}, \tag{11}$$

where $N_{\max}{=}\frac{\max_{m\in\boldsymbol{\mathcal{M}}}\{C_m\}}{\min_{n\in\boldsymbol{\mathcal{N}}}\{c_n\}}$, *and* $\delta_{\min}{=}\min\{\delta^1_{\min}, \delta^2_{\min}, \delta^3_{\min}\}$ *in which* $\delta^1_{\min} = \min_{n\neq n'\in\boldsymbol{\mathcal{N}}, m\in\boldsymbol{\mathcal{M}}}\{|u_{n,m} - u_{n',m}|\}$, $\delta^2_{\min} = \min_{n\neq n'\in\boldsymbol{\mathcal{N}}, m\neq m'\in\boldsymbol{\mathcal{M}}}\{|(u_{n,m} - u_{n,m'}) - u_{n',m}|\}, \delta^3_{\min} = \min_{n\neq n'\in\boldsymbol{\mathcal{N}}, m'\neq m\neq m''\in\boldsymbol{\mathcal{M}}}\{|(u_{n,m}{-}u_{n,m'}){-}(u_{n',m}{-}u_{n',m''})|\}$.

Proof. See Appendix B, via Hoeffding inequality [50]. □

Lemma 2 (Guaranteed Mapping Accuracy)**.** *If it holds that*

$$|\tilde{u}^{\tau}_{n,m} - u_{n,m}| \le \delta_{\min}/(5N_{\max}), \tag{12}$$

*then through **Alg. 2**, the mapping result* $\boldsymbol{\Pi}$ *derived from the observed reward estimates* $\{\tilde{u}^{\tau}_{n,m}\}$ *is the **same** as that from the expected rewards* $\{u_{n,m}, \forall n\in\boldsymbol{\mathcal{N}}, m\in\boldsymbol{\mathcal{M}}\}$.

Proof. See Appendix C, with the proof by contradiction. □

Lemma 3 (Approximation Ratio for Multi-Knapsack)**.** *With the optimal solution to the knapsack problem $\mathbb{P}_3$ upon branch-and-bound method, **Alg. 2** can obtain a **2-approximate** solution to the multi-knapsack problem $\mathbb{P}_2$.*

Proof. See Appendix D, via mathematical induction [48]. □

Theorem 1 (Bounded Regret)**.** *When **Alg. 1** is adopted to solve problem $\mathbb{P}_1$ without certain information of expected rewards $\{u_{n,m}, \forall n\in\boldsymbol{\mathcal{N}}, m\in\boldsymbol{\mathcal{M}}\}$, the regret is bounded as*

$$\mathcal{R}(T) \le (T_{explore} + M)Nu^{\max}\log_2(T+2) + 8N^2Mu^{\max}$$
$$= \mathcal{O}(\log_2 T).$$

Proof. See Appendix E, combining Lemmas 1, 2 and 3. □

Theorem 2 (Bounded Regret for Flexibility)**.** *For the case of leaving, through **Alg. 1**, the regret is still bounded as $\mathcal{R}(T) \le \mathcal{O}(\log_2 T)$. For the case of joining, denote the epoch of the last platform entering as τ', the regret is bounded as*

$$\mathcal{R}(T) \le \mathcal{O}(\log_2 T) + 2^{\mathcal{O}(\log_2 T^{\xi})}Mu^{\max} \le \mathcal{O}(T^{\xi}),$$

where $\xi \in (0,1)$ ensures inequation $\tau' \le \mathcal{O}(\log_2 T^{\xi})$ holds.

Proof. See Appendix F, combining Lemmas 1, 2 and 3. □

TABLE III
Description for Testbed, Input Dataset and Analytics Model

Content	Description
3 * PowerEdge R740	Silver 4210R 2.4G, 2 * 16GB RDIMM
3 * Raspberry Pi 4	ARM Cortex-A72 1.5GHz, 2GB RAM
Sales Product Dataset [51]	Sales data involving price, used as task values
PANDA Dataset [38]	Human-centric videos with wide field of view
YOLOv5n/s/x [39]	A family of compound-scaled detection models

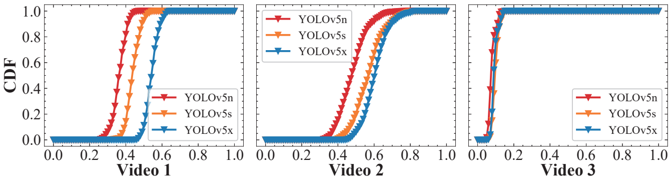


Fig. 6. CDFs of video analytics accuracy in our dataset

Remarks. Lemma 1 states that after a specified number of epochs for exploration, the estimated reward $\tilde{u}^{\tau}_{n,m}$ becomes sufficiently close to the expected reward $u_{n,m}$ with a high probability. Note that if $\delta_{\min}$ is too small and then leads to a prohibitively long exploration length T_{explore} as the input to **Alg. 1**, we can adjust $\delta_{\min}$ to be large enough in practice since T_{explore} in **Lemma 1** simply ensures an upper bound on exploration error. The logarithmic regret bound we derive in **Theorem 1** is tight since a lower $\log T$ regret bound can be deduced in a centralized way [25]. Besides, the reason for limiting τ' in **Theorem 2** is to prevent a large exploitation regret caused by inaccurate reward estimates of newly joining platforms. Finally, when platform n can only reach a subset of workers $\boldsymbol{\mathcal{M}}_n{\in}\boldsymbol{\mathcal{M}}$, the above regret analysis keeps unchanged as the rewards on $\boldsymbol{\mathcal{M}}_n$ can still be learned for sufficient times.

V. Experiments and Result Analysis

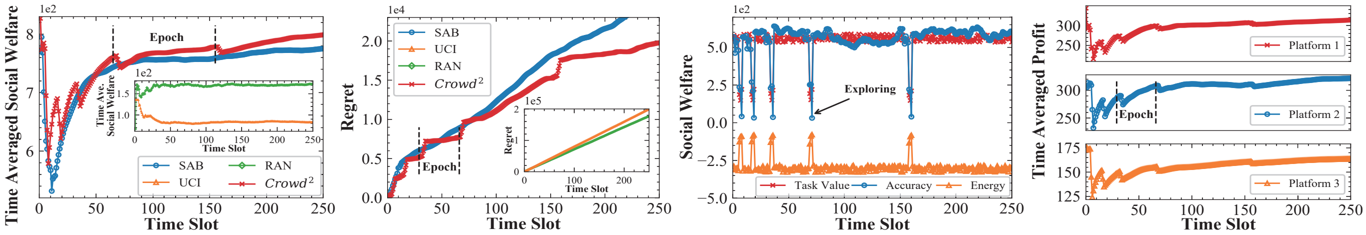
In this section, we demonstrate the superior performance of our proposed algorithm compared with other alternatives by both testbed-based experiments and scalable simulations.

A. Experiment Settings

Our testbed-based experiments are conducted on the PowerEdge R740s and Raspberry Pi 4s, which act as platforms and workers, respectively. For platforms, real video datasets [38] are used for video analytics tasks dispatched to workers, each equipped with a different video analytics model [39], as shown in Table III and Fig. 6. Based on the traces from [17, 32], the transmission energy consumption γ_n is set as $5{\times}10^{-6}$ J, and energy consumption of local processing is uniformly 5 J per frame. We generate the intrinsic task value $V(n)$ upon the dataset of Sales Product [51]. Besides, we set the parameters $\alpha{=}1$, G_n as the logarithmic form and $\omega_m \sim U(0,10)$. Furthermore, according to [22, 23, 52], the computation demand c_n is set in $[0.5, 1]$, and the capacity C_m is in $[1.5, 2]$.

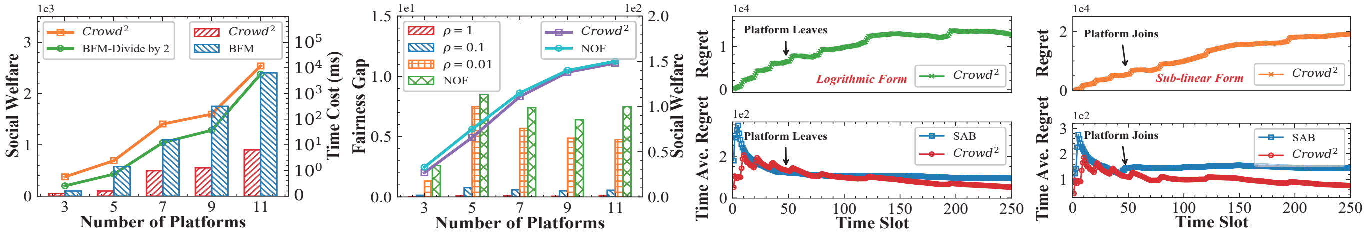
We compare our proposed algorithm with other schemes:

- **Single-Agent Bandit (SAB)** [32] designs, for each platform, a multi-armed bandit-based dispatch method using upper confidence bound (UCB) for video analytics.
- **NOt considering Fairness (NOF)** [21] aims to maximize the social welfare, i.e., the utilitarian compound reward in $\mathbb{P}$, not considering the proportional fairness.



(a) Comparing social welfare (b) Comparing regret (c) 3 components of social welfare (d) Profit for each platform

Fig. 7. Running instance on testbed with real-world dataset



(a) Effectiveness & efficiency of **Alg. 2** (b) Fairness for $Crowd^2$ (c) Flexibility: case of leaving (d) Flexibility: case of joining

Fig. 8. Evaluations for the social welfare, fairness and flexibility in mechanism $Crowd^2$

- **Unsolved Conflicts of Interest (UCI)** decides the worker selection based on their performance for video analytics, regardless of the conflicts of interest among platforms.
- **BruteForce for Multi-knapsack (BFM)** solves, as an alternative to **Alg. 2**, $\mathbb{P}_2$ of multi-knapsack upon bruteforce.
- **RANdomized dispatch for platforms (RAN)** dispatches video analytics tasks to workers randomly for platforms.

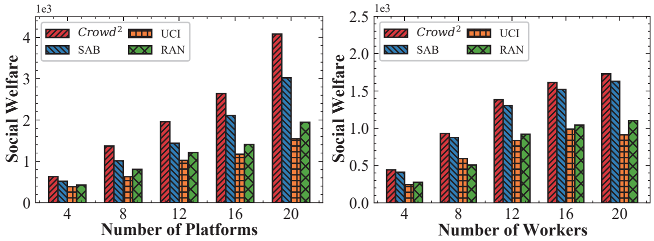
B. Experiment Results

Running Instance on Testbed. We demonstrate a small-scale running instance based on testbed with 3 platforms and 3 workers to verify the performance of $Crowd^2$. For video analytics, the platform first divides its video data into multiple chunks, each of which lasts for 1 second, and then dispatches one chunk to the recruited worker in each time slot. As shown in Fig. 7(a), the time-averaged reward from $Crowd^2$ is more than other algorithms, especially after the first 100 time slots, where more and more accurate profits estimates are obtained for the multi-agent bandit. SAB based on multi-armed bandit may tradeoff between exploration and exploitation, and then recruit unfamiliar and incompetent workers. Further, it cannot effectively learn the reduction in the performance of workers caused by the conflicts of interest, thus lowering the rewards of social welfare, which, however, can be avoided by the phases of exploration and mapping in $Crowd^2$. Besides, since RAN randomly selects workers for video analytics, it may recruit the workers leading to lower profits. Nevertheless, RAN achieves higher social welfare than UCI, which tends to select workers with higher performance, resulting in lower social welfare due to conflicts of interest. As shown in Fig. 7(b), our $Crowd^2$ achieves the lowest regret that satisfies the logarithmic bound with respect to slot number T, and outperforms other schemes. To dive into the social welfare changes by $Crowd^2$, we illustrate the 3 components of social welfare and the profit for each platform in Fig. 7(c) and Fig. 7(d), respectively. When each epoch begins, the three components of profits including the intrinsic task value, video analytics benefit and energy consumption cost will greatly fluctuate due to the exploration and mapping. Besides, as the number of epochs increases, the time-averaged reward for each platform also gradually rises.

Approximation for Mapping. We compare our proposed **Alg. 2** with BFM, which solves the multiple knapsack problem by means of brute force, as shown in Fig. 8(a). To conveniently demonstrate the effectiveness of **Alg. 2**, we show the half values of optimal social welfare obtained by BFM, and it can be observed that our proposed algorithm achieves the 2-approximate solutions regardless of the number of platforms. Besides, we compare the time cost by **Alg. 2** and BFM. When the number of platforms increases, the computational time of BFM grows exponentially while the time cost caused by our proposed algorithm of mapping upon multi-knapsack remains stable, which reflects that our mechanism $Crowd^2$ is efficient.

Fairness for $Crowd^2$. To confirm the fairness guaranteed by $Crowd^2$, we compare our proposed algorithm with NOF, which simply maximizes the utilitarian compound reward. As shown in Fig. 8(b), we find our proportional fairness-based profit maximization, with the parameter ρ, greatly reduces the fairness gap, which is represented as the profit variance [27]. In addition, we study the impact of parameter ρ with different values on fairness. As parameter ρ increases, the fairness can be further guaranteed due to the fact that the gradient of the logarithmic function with larger ρ will get smaller, which leads to the difference among platforms shrinking. Furthermore, we explore the effect of introducing fairness guarantees on social welfare, and it can be noticed that when the parameter ρ is set as 0.01, at the cost of sacrificing at most 7.5% crowdsourcing welfare, we reduce the fairness gap by 25.6%.

Flexibility for $Crowd^2$. For the case of leaving, as shown in Fig. 8(c), one platform exits at about the 50th time slot. With the platform leaving, since the remaining platforms in $Crowd^2$ can still leverage the learned information to schedule subsequent video analysis tasks, the regret values are kept in logarithmic forms, consistent with **Theorem 2**, as shown in the upper sub-figure of Fig. 8(c). Besides, compared with SAB,



(a) Number of platforms (b) Number of workers

Fig. 9. Scalability on number of platforms/workers

which learns the worker performance via single-agent bandit, $Crowd^2$ achieves lower time-averaged regret, especially after the 100th time slot. For the case of joining, as illustrated in Fig. 8(d), there is one platform joining at the 50th time slot. Our proposed $Crowd^2$ and SAB both need to spend some time slots to explore the workers for the new platform, which has an impact on the regret. Regarding the regret changes shown in the upper sub-figure of Fig. 8(d), the regret bound of $Crowd^2$, consistent with theoretical analysis, remains in a sub-linear form. Furthermore, due to the joining of the new platform, the time-averaged regret of SAB experiences a sudden rise at about the 50th time slot, which is higher than $Crowd^2$.

Scalability on platform/worker number. We finally evaluate the scalability of our proposed $Crowd^2$, when changing the number of platforms and workers. As shown in Fig. 9, we vary the numbers of platforms and workers from 4 to 20, and find that our proposed algorithm can achieve higher social welfare than other algorithms, which reflects the scalability of $Crowd^2$. Concretely, compared with others, our proposed algorithm averagely improves 36.8% social welfare.

VI. Related Work

We summarize the prior existing works by the following categories and then highlight their defects compared with ours.

A. Worker Recruitment for Crowdsourcing

Liu *et al.* [30] studied the sparse crowdsensing considering online workers with dynamically coming data and proposed the OS-MCS framework consisting of matrix completion, importance estimation and worker recruitment. Wang *et al.* [31] designed a privacy-preserving online task assignment framework to minimize the total travel distance with the assigned task cardinality constraint. Chen *et al.* [32] formulated a mixed integer program maximizing the crowdsourcing profit to determine the most suitable workers upon Lyapunov optimization and volatile multi-armed bandit. Liu *et al.* [33] proposed a dynamic worker recruitment strategy with truthful pricing for online recruitment problems constrained by budget and time.

These works focus on determining worker selection to maximize the crowdsourcing utility in various scenarios. However, almost no work considers the social welfare for the crowdsourcing platforms which collaboratively recruit workers in a decentralized way, and it is covered in our $Crowd^2$.

B. Configuration for Video Analytics

DDS [10] leveraged the compression and streaming behaviors driven by the feedback from server-side DNNs rather than the low-complexity heuristics from cameras to reduce bandwidth usage while maintaining higher accuracy. Reducto [11] was built as a system adapting filtering decisions dynamically based on the time-varying correlations between video content, feature type, filtering threshold and query accuracy. JCAB [12] was proposed as an efficient online algorithm optimizing the configuration adaption and bandwidth allocation in edge-based video analytics systems. Chameleon [13] used several techniques including exploiting the independence of configuration knobs, temporal persistence of configurations and cross-video similarities to dramatically reduce cost and improve accuracy.

These works consider the configuration adaptation for video analytics, but fail to consider the impact of video analytics on energy consumption and crowdsourcing profits.

C. Multi-Armed Bandit-based Optimization

Xiong *et al.* [34] proposed a fluid Whittle index policy determining dimensioning and content caching to tackle the restless multi-armed bandit problem of minimizing the costs with respect to storage and latency. Gao *et al.* [35] considered the unknown worker recruitment problem and proposed a combinatorial multi-armed bandit-based worker recruitment algorithm. Song *et al.* [36] designed a novel worker recruitment mechanism for minimum empirical entropy of the results from participating workers by means of combinatorial multi-armed bandit. Yang *et al.* [37] investigated contextual bandit with predicted context and proposed selective context query algorithms for more accurate context subject to query budget.

These works leverage multi-armed bandit to tackle the optimization problem in different scenarios. However, they fail to consider multiple agents to explore and exploit the shared resources while taking into account fairness and flexibility.

VII. Conclusion

To maximize the social welfare for crowdsourcing platforms dispatching video analytics tasks, we propose a decentralized mechanism $Crowd^2$, considering both fairness and flexibility. For maximum crowdsourcing profits depending on video analytics accuracy and energy consumption, which are uncertain and time-varying, we optimize the worker selection based on multi-agent multi-armed bandit in long term, decoupled into multiple knapsack sub-problems for each worker. Via rigorous proof, a sub-linear regret for profits is guaranteed. Extensive trace-driven experiments show the superiority of our proposed mechanism $Crowd^2$ compared with other existing works.

Appendix

A. Proof of Proposition 1

Proof. Since $u_{n,x_{n,t}}(r_{n,x_{n,t}}) = \ln(1 + \rho r_{n,x_{n,t}})$ is a concave function, it is satisfied that for any other decision $x'_{n,t}$,

$$\nabla u_{n,x^*_{n,t}}(r_{n,x^*_{n,t}})(r_{n,x'_{n,t}} - r_{n,x^*_{n,t}}) \leq 0.$$

Therefore, according to [45], for the optimal solution $\boldsymbol{x}^*_t$, it always holds that $\sum_{n\in\mathcal{N}} \frac{r_{n,x'_{n,t}} - r_{n,x^*_{n,t}}}{r_{n,x^*_{n,t}}} \leq 0$. □

B. Proof of Lemma 1

Proof. Denote event $\mathcal{A}$ as $\{\exists n\in\boldsymbol{\mathcal{N}}, m\in\boldsymbol{\mathcal{M}}, |\tilde{u}^{\tau}_{n,m}-u_{n,m}|>\frac{\delta_{\min}}{5N_{\max}}\}$. After the τ-th epoch of exploration in **Alg. 1**, each platform n observes the reward from worker m at least $T_{\min}\geq\frac{T_{\text{explore}}}{M}\tau\geq\frac{25N^2_{\max}\left(u^{\max}-u^{\min}\right)^2}{2(\delta_{\min})^2}\tau$ times. Therefore, the exploration error probability can be calculated as

$$\begin{aligned}\Pr(\mathcal{A}|T_{\min})&\leq\sum\nolimits_{n\in\boldsymbol{\mathcal{N}}}\sum\nolimits_{m\in\boldsymbol{\mathcal{M}}}\Pr(|\tilde{u}^{\tau}_{n,m}-u_{n,m}|>\frac{\delta_{\min}}{5N_{\max}})\\&\leq NM\max_{n\in\boldsymbol{\mathcal{N}},m\in\boldsymbol{\mathcal{M}}}\Pr(|\tilde{u}^{\tau}_{n,m}-u_{n,m}|>\frac{\delta_{\min}}{5N_{\max}})\\&\overset{(a)}{\leq}2NMe^{-\frac{2(\delta_{\min})^2T_{\min}}{25N^2_{\max}(u^{\max}-u^{\min})^2}}\leq 2NMe^{-\tau},\end{aligned}$$

where inequation (a) is from Hoeffding inequality [50]. □

C. Proof of Lemma 2

Proof. According to **Lemma 1**, it always holds that

$$\Pr(|\tilde{u}^{\tau}_{n,m}-u_{n,m}|\leq\delta_{\min}/(5N_{\max}))>1-2NMe^{-\tau},$$

for the mapping phase running in **Alg. 2**, and then we define $\delta_{n,m}=\tilde{u}^{\tau}_{n,m}-u_{n,m}$. Besides, denote the mapping result under expected rewards $\{u_{n,m},\forall n\in\boldsymbol{\mathcal{N}},m\in\boldsymbol{\mathcal{M}}\}$ as $\overline{\boldsymbol{\Pi}}=\{\overline{\Pi}_n,n\in\boldsymbol{\mathcal{N}}\}$ and the decision induced from $\overline{\boldsymbol{\Pi}}$ as $\overline{\boldsymbol{x}}=\{\overline{\boldsymbol{x}}_t,t\in\boldsymbol{\mathcal{T}}\}$. Then, we discuss the knapsack sub-problem corresponding to worker m under the updating mapping $\overline{\boldsymbol{\Pi}}$ and $\boldsymbol{\Pi}$, separately.

For the updating mapping $\overline{\boldsymbol{\Pi}}$, we define

$$\text{Opt}_m(\overline{\boldsymbol{x}})=\sum\nolimits_{n\in\boldsymbol{\mathcal{N}}}\Delta u_{n,m}=\sum\nolimits_{n\in\boldsymbol{\mathcal{N}}}u_{n,m}-u_{n,\overline{\Pi}_n}$$

and

$$\widetilde{\text{Opt}}_m(\overline{\boldsymbol{x}})=\sum\nolimits_{n\in\boldsymbol{\mathcal{N}}}\Delta\tilde{u}^{\tau}_{n,m}=\sum\nolimits_{n\in\boldsymbol{\mathcal{N}}}\tilde{u}^{\tau}_{n,m}-\tilde{u}^{\tau}_{n,\overline{\Pi}_n}.$$

Then, with the probability of more than $1-2NMe^{-\tau}$, we have

$$\begin{aligned}\text{Opt}_m(\overline{\boldsymbol{x}})-\widetilde{\text{Opt}}_m(\overline{\boldsymbol{x}})&\leq N_{\max}\max\{|\delta_{n,m}|+|\delta_{n,\bar{\Pi}_n}|\}\\&\leq N_{\max}\frac{2\delta_{\min}}{5N_{\max}}\leq 2\delta_{\min}/5.\end{aligned}$$

Similarly, for the updating mapping $\boldsymbol{\Pi}$ and the decision $\boldsymbol{x}=\{\boldsymbol{x}_t,t\in\boldsymbol{\mathcal{T}}\}$ induced from $\boldsymbol{\Pi}$, we define

$$\text{Opt}_m(\boldsymbol{x})=\sum\nolimits_{n\in\boldsymbol{\mathcal{N}}}\Delta u_{n,m}=\sum\nolimits_{n\in\boldsymbol{\mathcal{N}}}u_{n,m}-u_{n,\Pi_n}$$

and

$$\widetilde{\text{Opt}}_m(\boldsymbol{x})=\sum\nolimits_{n\in\boldsymbol{\mathcal{N}}}\Delta\tilde{u}^{\tau}_{n,m}=\sum\nolimits_{n\in\boldsymbol{\mathcal{N}}}\tilde{u}^{\tau}_{n,m}-\tilde{u}^{\tau}_{n,\Pi_n}.$$

Then we can also obtain $\widetilde{\text{Opt}}_m-\text{Opt}_m(\boldsymbol{x})\leq 2\delta_{\min}/5$.

According to the above inequations, we have

$$\begin{aligned}&\text{Opt}_m(\overline{\boldsymbol{x}})-\text{Opt}_m(\boldsymbol{x})\\&=\text{Opt}_m(\overline{\boldsymbol{x}})-\widetilde{\text{Opt}}_m(\overline{\boldsymbol{x}})+\widetilde{\text{Opt}}_m(\overline{\boldsymbol{x}})-\text{Opt}_m(\boldsymbol{x})\\&\overset{(b)}{\leq}\text{Opt}_m(\overline{\boldsymbol{x}})-\widetilde{\text{Opt}}_m(\overline{\boldsymbol{x}})+\widetilde{\text{Opt}}_m(\boldsymbol{x})-\text{Opt}_m(\boldsymbol{x})\overset{(c)}{\leq}4\delta_{\min}/5,\end{aligned}$$

where inequation (b) is in that $\widetilde{\text{Opt}}_m(\overline{\boldsymbol{x}})\leq\widetilde{\text{Opt}}_m(\boldsymbol{x})$ always holds under the input rewards $\{\Delta\tilde{u}^{\tau}_{n,m},n\in\boldsymbol{\mathcal{N}}\}$.

Besides, based on the definition of $\delta_{\min}$, we have

$$\text{Opt}_m(\overline{\boldsymbol{x}})-\text{Opt}_m(\boldsymbol{x})\geq\delta_{\min},\forall m\in\boldsymbol{M},$$

which is contradictory to inequation (c). Thus, it must hold that $\text{Opt}_m(\overline{\boldsymbol{x}})=\text{Opt}_m(\boldsymbol{x}),\forall m\in M$, which means $\boldsymbol{x}$ is actually $\overline{\boldsymbol{x}}$. Therefore, through **Alg. 2**, the mapping result $\boldsymbol{\Pi}$ derived from the estimated rewards $\{\tilde{u}^{\tau}_{n,m}\}$ is the **same** as that from the expected rewards $\{u_{n,m},\forall n\in\boldsymbol{\mathcal{N}},m\in\boldsymbol{\mathcal{M}}\}$. □

D. Proof of Lemma 3

Proof. Due to page limit, we prove **Lemma 3** based on the running example as shown in Figure 4, and more details can be found in [48]. The proof is presented by induction.

Base case. When the worker number is 1 (e.g., worker m_3), the optimal solution to knapsack sub-problem is indeed a 2-approximate solution to multi-knapsack sub-problem.

Inductive step. We then consider the worker number of 2 (e.g., worker m_2 and m_3). According to the base case, the derived mapping upon matrix U_3 is 2-approximate, and it is also true for U_2^2, which only has one more row of 0 than U_3.

On the other hand, we separately discuss the reward gains resulting from the three parts in U_2^1. For the 1st part, the optimal solution S_{m_2} of knapsack sub-problem for worker m_2 is optimal to part 1 in U_2^1 as well. For the 2nd part, the optimal reward gain from part 2 in U_2^1 will not exceed that of part 1 due to the fact that the columns covered in S_{m_2} are the same as those in part 2. The remaining part 3 will not contribute to the reward gain since it is filled with 0. Furthermore, S_{m_2} is the subset of the derived mapping, and thus the derived mapping upon U_2^1 is also a 2-approximate solution.

To sum up, the derived mapping upon both U_2^1 and U_2^2 is 2-approximate. Besides, we have $U_2=U_2^1+U_2^2$. Therefore, the derived mapping upon U_2 is also 2-approximate.

Conclusion. By induction, the derived mapping upon U_1 is also 2-approximate, which means that the approximation ratio for multi-knapsack problem through **Alg. 2** is 2. □

E. Proof of Theorem 1

Proof. We first bound τ_T using T. Since the exploiting phase occupies 2^{τ} time slots for epoch τ in **Alg. 1**, we have $T\geq\sum_{\tau=1}^{\tau_T}2^{\tau}=2(2^{\tau_T}-1)$, which means $\tau_T\leq\log(T+2)$, and it still holds when we replace $T_{\text{exploit}}=2^{\tau}$ with other exponential forms (e.g., 3^{τ}). We represent the regret for $\mathbb{P}_2$ as

$$\begin{aligned}\mathcal{R}(T)&\overset{(d)}{\leq}\sum_{\tau=1}^{\tau_T}(T_{\text{explore}}Nu^{\max}+MNu^{\max}+2NMe^{-\tau}T_{\text{exploit}}Nu^{\max})\\&\overset{(e)}{\leq}(T_{\text{explore}}Nu^{\max}+MNu^{\max})\tau_T+4N^2Mu^{\max}/(e-2)\\&\leq(T_{\text{explore}}Nu^{\max}+MNu^{\max})\log(T+2)+8N^2Mu^{\max}\\&=\mathcal{O}(\log_2 T),\end{aligned}$$

where inequation (d) represents the sum of regrets from the 3 phases in **Alg. 1**, and (e) combines Lemmas 1-3. □

F. Proof of Theorem 2

Proof. For the case of leaving, we denote the platform number in epoch τ as N_{τ}. After some platform leaves, we have $N_{\tau}\leq N$, and then the exploration error probability $\mathcal{P}$ satisfies $\mathcal{P}\leq 2N_{\tau}Me^{-\tau}\leq 2NMe^{-\tau}$, thus leading to a regret of $\mathcal{O}(\log_2 T)$.

For the case of joining, denote τ' as the epoch of the last platform entering, and it holds that $\tau'\leq\mathcal{O}(\log_2 T^{\xi}),\xi\in(0,1)$. Thus, we have $\forall\tau\geq\tau',\mathcal{P}\leq 2N_{\tau}Me^{-(\tau+1-\tau')}$. Furthermore, there must exist $\xi\in(0,1),\tau_0\in[\tau',\mathcal{O}(\log_2 T^{\xi})]$ such that $\forall\tau\geq\tau_0,2N_{\tau}Me^{-(\tau+1-\tau')}\times 2^{\tau}\sim\mathcal{O}(1)$. Therefore, we get the upper bounded regret for the case of joining as

$$\begin{aligned}\mathcal{R}(T)&\overset{(f)}{\leq}2\times\mathcal{O}(\log_2 T)+\sum_{\tau=1}^{\tau_T}Mu^{\max}\times 2^{\tau}\times 2N_{\tau}Me^{-(\tau+1-\tau')}\\&\overset{(g)}{\leq}\mathcal{O}(\log_2 T)+\sum\nolimits_{\tau=1}^{\tau_0-1}Mu^{\max}\times 2^{\tau}+\sum\nolimits_{\tau=\tau_0}^{\tau_T}\mathcal{O}(1)\\&\leq\mathcal{O}(\log_2 T)+Mu^{\max}\times 2^{\mathcal{O}(\log_2 T^{\xi})}+\mathcal{O}(\log_2 T)\leq\mathcal{O}(T^{\xi}),\end{aligned}$$

where inequation (f) represents the sum of regrets from the 3 phases in **Alg. 1**, and upon Lemmas 1-3, (g) separately considers the exploitation regret before and after epoch τ_0. □

References

[1] W. Jin, M. Xiao, M. Li, and L. Guo, "If you do not care about it, sell it: Trading location privacy in mobile crowd sensing," in *IEEE INFOCOM*, 2019, pp. 1045–1053.

[2] X. Kong, X. Liu, B. Jedari, M. Li, L. Wan, and F. Xia, "Mobile crowdsourcing in smart cities: Technologies, applications, and future challenges," *IEEE IoTJ*, vol. 6, no. 5, pp. 8095–8113, 2019.

[3] Y. Tian, W. Wei, Q. Li, F. Xu, and S. Zhong, "Mobicrowd: Mobile crowdsourcing on location-based social networks," in *IEEE INFOCOM*, 2018, pp. 2726–2734.

[4] Z. Wang, J. Li, J. Hu, J. Ren, Z. Li, and Y. Li, "Towards privacy-preserving incentive for mobile crowdsensing under an untrusted platform," in *IEEE INFOCOM*, 2019, pp. 2053–2061.

[5] G. Paolacci, J. Chandler, and P. G. Ipeirotis, "Running experiments on amazon mechanical turk," *Judgment and Decision making*, vol. 5, no. 5, pp. 411–419, 2010.

[6] C. Van Pelt and A. Sorokin, "Designing a scalable crowdsourcing platform," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, 2012, pp. 765–766.

[7] D. R. Karger, S. Oh, and D. Shah, "Efficient crowdsourcing for multi-class labeling," in *ACM SIGMETRICS*, 2013, pp. 81–92.

[8] S. Bianco, G. Ciocca, P. Napoletano, and R. Schettini, "An interactive tool for manual, semi-automatic and automatic video annotation," *Elsevier CVIU*, vol. 131, pp. 88–99, 2015.

[9] D. Oleson, A. Sorokin, G. Laughlin, V. Hester, J. Le, and L. Biewald, "Programmatic gold: Targeted and scalable quality assurance in crowdsourcing," in *AAAI Workshop*, 2011.

[10] K. Du, A. Pervaiz, X. Yuan, A. Chowdhery, Q. Zhang, H. Hoffmann, and J. Jiang, "Server-driven video streaming for deep learning inference," in *ACM SIGCOMM*, 2020, pp. 557–570.

[11] Y. Li, A. Padmanabhan, P. Zhao, Y. Wang, G. H. Xu, and R. Netravali, "Reducto: On-camera filtering for resource-efficient real-time video analytics," in *ACM SIGCOMM*, 2020, pp. 359–376.

[12] C. Wang, S. Zhang, Y. Chen, Z. Qian, J. Wu, and M. Xiao, "Joint configuration adaptation and bandwidth allocation for edge-based real-time video analytics," in *IEEE INFOCOM*, 2020, pp. 257–266.

[13] J. Jiang, G. Ananthanarayanan, P. Bodik, S. Sen, and I. Stoica, "Chameleon: scalable adaptation of video analytics," in *ACM SIGCOMM*, 2018, pp. 253–266.

[14] Z. Dai, C. H. Liu, Y. Ye, R. Han, Y. Yuan, G. Wang, and J. Tang, "Aoi-minimal uav crowdsensing by model-based graph convolutional reinforcement learning," in *IEEE INFOCOM*, 2022, pp. 1029–1038.

[15] C. H. Liu, C. Piao, and J. Tang, "Energy-efficient uav crowdsensing with multiple charging stations by deep learning," in *IEEE INFOCOM*, 2020, pp. 199–208.

[16] Y. Chen, S. Zhang, M. Xiao, Z. Qian, J. Wu, and S. Lu, "Multi-user edge-assisted video analytics task offloading game based on deep reinforcement learning," in *IEEE ICPADS*, 2020, pp. 266–273.

[17] H. Qian and D. Andresen, "Reducing mobile device energy consumption with computation offloading," in *IEEE SNPD*, 2015, pp. 1–8.

[18] H. Li and L. Chen, "Rssi-aware energy saving for large file downloading on smartphones," *IEEE LES*, vol. 7, no. 2, pp. 63–66, 2015.

[19] Z. Wang, Y. Huang, X. Wang, J. Ren, Q. Wang, and L. Wu, "Socialrecruiter: Dynamic incentive mechanism for mobile crowdsourcing worker recruitment with social networks," *IEEE TMC*, vol. 20, no. 5, pp. 2055–2066, 2020.

[20] C. H. Liu, Z. Dai, H. Yang, and J. Tang, "Multi-task-oriented vehicular crowdsensing: A deep learning approach," in *IEEE INFOCOM*, 2020, pp. 1123–1132.

[21] H. Jin, H. Guo, L. Su, K. Nahrstedt, and X. Wang, "Dynamic task pricing in multi-requester mobile crowd sensing with markov correlated equilibrium," in *IEEE INFOCOM*, 2019, pp. 1063–1071.

[22] Y. Wu, Y. Wang, and G. Cao, "Photo crowdsourcing for area coverage in resource constrained environments," in *IEEE INFOCOM*, 2017, pp. 1–9.

[23] Y. Jararweh, A. Doulat, O. AlQudah, E. Ahmed, M. Al-Ayyoub, and E. Benkhelifa, "The future of mobile cloud computing: integrating cloudlets and mobile edge computing," in *IEEE ICT*, 2016, pp. 1–5.

[24] H. Jin, L. Su, and K. Nahrstedt, "Centurion: Incentivizing multi-requester mobile crowd sensing," in *IEEE INFOCOM*, 2017, pp. 1–9.

[25] I. Bistritz, T. Baharav, A. Leshem, and N. Bambos, "My fair bandit: Distributed learning of max-min fairness with multi-player bandits," in *PMLR ICML*, 2020, pp. 930–940.

[26] Z. Shi, S. Jiang, L. Zhang, Y. Du, and X.-Y. Li, "Crowdsourcing system for numerical tasks based on latent topic aware worker reliability," in *IEEE INFOCOM*, 2021, pp. 1–10.

[27] L. Li, M. Pal, and Y. R. Yang, "Proportional fairness in multi-rate wireless lans," in *IEEE INFOCOM*, 2008, pp. 1004–1012.

[28] Y. Li, H. Sun, and W. H. Wang, "Towards fair truth discovery from biased crowdsourced answers," in *ACM SIGKDD*, 2020, pp. 599–607.

[29] X. Wang, R. Jia, X. Tian, and X. Gan, "Dynamic task assignment in crowdsensing with location awareness and location diversity," in *IEEE INFOCOM*, 2018, pp. 2420–2428.

[30] W. Liu, E. Wang, Y. Yang, and J. Wu, "Worker selection towards data completion for online sparse crowdsensing," in *IEEE INFOCOM*, vol. 33, 2022.

[31] H. Wang, E. Wang, Y. Yang, J. Wu, and F. Dressler, "Privacy-preserving online task assignment in spatial crowdsourcing: A graph-based approach," in *IEEE INFOCOM*, 2022, pp. 570–579.

[32] Y. Chen, S. Zhang, Y. Jin, Z. Qian, M. Xiao, N. Chen, and Z. Ma, "Learning for crowdsourcing: Online dispatch for video analytics with guarantee," in *IEEE INFOCOM*, 2022, pp. 1908–1917.

[33] W. Liu, Y. Yang, E. Wang, and J. Wu, "Dynamic user recruitment with truthful pricing for mobile crowdsensing," in *IEEE INFOCOM*, 2020, pp. 1113–1122.

[34] G. Xiong, S. Wang, G. Yan, and J. Li, "Reinforcement learning for dynamic dimensioning of cloud caches: A restless bandit approach," in *IEEE INFOCOM*, 2022, pp. 2108–2117.

[35] G. Gao, J. Wu, M. Xiao, and G. Chen, "Combinatorial multi-armed bandit based unknown worker recruitment in heterogeneous crowdsensing," in *IEEE INFOCOM*, 2020, pp. 179–188.

[36] Y. Song and H. Jin, "Minimizing entropy for crowdsourcing with combinatorial multi-armed bandit," in *IEEE INFOCOM*, 2021, pp. 1–10.

[37] J. Yang and S. Ren, "Bandit learning with predicted context: Regret analysis and selective context query," in *IEEE INFOCOM*, 2021, pp. 1–10.

[38] X. Wang, X. Zhang, Y. Zhu, Y. Guo, X. Yuan, L. Xiang, Z. Wang, G. Ding, D. J. Brady, Q. Dai, and L. Fang, "Panda: A gigapixel-level human-centric video dataset," in *CVPR*. IEEE, 2020.

[39] G. J. *et al.*, "ultralytics/yolov5: v3.0," Aug. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3983579

[40] Q. Liu and T. Han, "Dare: Dynamic adaptive mobile augmented reality with edge computing," in *IEEE ICNP*, 2018, pp. 1–11.

[41] Z. Lu, S. Rallapalli, K. Chan, and T. La Porta, "Modeling the resource requirements of convolutional neural networks on mobile devices," in *ACM MM*, 2017, pp. 1663–1671.

[42] P. Lai, Q. He, G. Cui, F. Chen, M. Abdelrazek, J. Grundy, J. Hosking, and Y. Yang, "Quality of experience-aware user allocation in edge computing systems: A potential game," in *IEEE ICDCS*, 2020, pp. 223–233.

[43] C. Zhou, C.-K. Tham, and M. Motani, "Online auction for scheduling concurrent delay tolerant tasks in crowdsourcing systems," *Elsevier CN*, vol. 169, p. 107045, 2020.

[44] T. X. Tran, K. Chan, and D. Pompili, "Costa: Cost-aware service caching and task offloading assignment in mobile-edge computing," in *IEEE SECON*, 2019, pp. 1–9.

[45] R. Srikant and L. Ying, *Communication networks: an optimization, control, and stochastic networks perspective*. Cambridge University Press, 2013.

[46] A. R. Kan, L. Stougie, and C. Vercellis, "A class of generalized greedy algorithms for the multi-knapsack problem," *Discrete applied mathematics*, vol. 42, no. 2-3, pp. 279–290, 1993.

[47] K. Pak and R. Dekker, "Cargo revenue management: Bid-prices for a 0-1 multi knapsack problem," *Available at SSRN 594991*, 2004.

[48] R. Cohen, L. Katzir, and D. Raz, "An efficient approximation for the generalized assignment problem," *Information Processing Letters*, vol. 100, no. 4, pp. 162–166, 2006.

[49] R. Neapolitan and K. Naimipour, "Foundations of algorithms using java pseudo code, jones and bartleet publishers," ISBN-978-443-5000,, Tech. Rep., 2004.

[50] J. Duchi, "Probability bounds," 2009. [Online]. Available: https://stanford.edu/∼jduchi/projects/probability_bounds.pdf

[51] "Sales Product Data," https://www.kaggle.com/datasets/knightbearr/sales-product-data, 2019.

[52] X. Wang, J. Ye, and J. C. Lui, "Decentralized task offloading in edge computing: A multi-user multi-armed bandit approach," in *IEEE INFOCOM*, 2022, pp. 1199–1208.