Risk-Aware Multi-Agent Multi-Armed Bandits

Qi Shao, Jiancheng Ye, and John C.S. Lui

ABSTRACT

Multi-armed bandits (MAB) is an online learning and decision-making model under uncertainty. Instead of maximizing the *expected* utility (or reward) in a classical MAB setting, the *variance* of the utility should be considered when making risk-aware decisions. In this paper, we propose a risk-aware multi-agent MAB (MAMAB) model, which considers both the "*independent*" and "*correlated*" risk when multiple agents make arm-pulling decisions. Specifically, the system includes a platform that owns a number of tasks (or arms) awaiting a group of agents to accomplish. We show how to calculate the arm-pulling strategy of agents with potentially different eligible arm sets under a Nash equilibrium point. From the perspective of the platform, each arm has its maximal capacity to accommodate arm-pulling agents. We design the platform's optimal payment algorithms for its risk-aware revenue maximization (a regret minimization) under both independent and correlated risks. We prove that our algorithms achieve the sub-linear regret under independent risks when the platform can or cannot differentiate the utility on each arm. We also prove that our algorithm achieves the sub-linear regret under correlated risks. We also carry out experiments to quantify the merits of our algorithms for various networking applications, such as crowdsourcing and edge computing.

CCS CONCEPTS

• **Theory of computation** → **Algorithmic game theory and mechanism design**; **Online learning algorithms**; **Regret bounds**; • **Computing methodologies** → **Multi-agent systems**.

KEYWORDS

Multi-armed bandits, multi-agent systems, risk-aware bandits

ACM Reference Format:

Qi Shao, Jiancheng Ye, and John C.S. Lui. 2024. Risk-Aware Multi-Agent Multi-Armed Bandits. In *International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '24), October 14–17, 2024, Athens, Greece.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3641512.3686368

Qi Shao (e-mail: qishao@cuhk.edu.hk) and John C.S. Lui (e-mail: cslui@cse.cuhk.edu.hk) are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong. Jiancheng Ye (e-mail: jcye@must.edu.mo) is with the School of Computer Science and Engineering, Macau University of Science and Technology. (Corresponding author: Jiancheng Ye.)

This work is supported in part by the RGC's GRF-14202923.

MobiHoc '24, October 14–17, 2024, Athens, Greece

ACM ISBN 979-8-4007-0521-2/24/10…$15.00
https://doi.org/10.1145/3641512.3686368

1 INTRODUCTION

1.1 Motivations

Stochastic multi-armed bandits (MAB) [1] is an online learning and decision-making model under uncertainty. The utility (or reward) of making each decision (or pulling each arm) is drawn from an unknown distribution. A classical stochastic MAB system involves an agent choosing an arm from a set of arms at each time step, and the agent receives its corresponding utility upon pulling the chosen arm. To maximize the total utility of pulling the arms, the agent needs to balance between learning the arms with high uncertainty (exploration) and choosing the arms with high empirical mean utility so far (exploitation). The performance of the arm-pulling policy is measured by *regret*, which is the average cumulative utility differences between the optimal arm and the selected arms.

Instead of maximizing the *expected* utility in a classical stochastic MAB setting, an agent may also be interested in reducing the risk (or uncertainty) when making decisions. A widely adopted measurement of risk (or uncertainty) is *mean-variance* (MV), which trades off between mean and variance. Bandits with MV provide a risk-aware model compared with the risk-neutral one in the classical setting. Nowadays, due to the extensive requirements of risk-aversion decisions, bandits with MV have received increasing attention [2–4]. However, bandits with MV only consider the setting that one agent only pulls one arm (with variances of individual arms) at each time.

In many real-world scenarios, we have multiple agents making their decisions simultaneously, which leads to "*correlated risks*". For example, multiple types of drugs (or arms) of a treatment may cause correlated effects and risks, or multiple signal transmissions choosing the same channel (or arm) may cause high delay or channel contention. Existing multi-agent MAB (MAMAB) studies usually focus on how agents cooperatively play the same MAB and achieve a common goal, i.e., the maximum total cumulative utility. The existing bandits model with MV [2–4], however, do not handle this MAMAB problem with correlated risks.

In this paper, we propose a risk-aware MAMAB model, which considers both the independent and correlated risk of decisions when multiple agents are involved. In the system, we consider a platform Foap [5] which owns multiple tasks (or arms) waiting for a number of agents to accomplish. The platform publishes multiple tasks (arms) of collecting photos with money rewards (payments). Each uploaded photo on one task generates a random utility for the platform. Based on the platform's payment, each "selfish" agent needs to choose and finish one specific task per time step. platform also gives out payment to those agents who select that arm. The platform aims to maximize its risk-aware revenue from the accomplished tasks, while agents aim to maximize its payoff (i.e., the difference between the payment and cost of selecting the tasks).

Given the payments set by the platform, each agent observes the arms on the platform and chooses one specific arm to pull (including not pulling any arm). Note that the arm-pulling decision of one agent depends not only on the payments on these arms, but also on other agents' arm-pulling decisions. For instance, if the payment on

one arm is high, many agents may select this high-payment arm. So choosing this arm may not a good choice for an agent. In this paper, we also consider agents with heterogeneous properties, such as the qualification of pulling arms. That is, certain arms are restricted to some important agents/persons (VIPs) of the platform. Only VIPs can access all arms, while non-VIPs can only access a subset of arms. This heterogeneity of agents makes the agents' arm-pulling decisions more realistic but challenging.

We focus on the risk-aware MAMAB system where each arm has its maximal utility (or capacity). This is practical for a wide range of applications, e.g., at a time step, a computing device can process a finite number of tasks, or a channel can only transmit a finite number of signals. We model this by setting the maximal utility of each arm be bounded by its maximal capacity. More specifically, on the platform Foap, the *effective* number of uploaded photos bounds the total utility for each task, as the utility does not increase infinitely with the number of collected photos. This effective number of uploaded photos is regarded as the capacity of each task, which is also unknown to the platform.

In the risk-aware MAMAB system, the platform can be of two different types when facing selected arms of same time step: 1) independent risks, and 2) correlated risks. The first type occurs when multiple agents pulling multiple arms will not generate significant correlated risks, e.g., multiple workers choose the same sensing tasks in a crowdsensing application. For this type, we first consider the case when the platform knows exactly the generated utility on each arm. Then the platform aims to optimize the aggregate of agents' generated cumulative MV of utility. Then we study the case when the platform cannot differentiate the arms' utilities, but only knows the aggregate of the utility, and the platform aims to optimize the cumulative MV of the utility summation. For the second type, considering the correlated risks when multiple agents pull multiple arms at the same time step, the platform optimizes the cumulative mean-covariance (MCV) of the utility. This scenario occurs with non-negligible correlated risks, i.e., different drugs in treatment, or different signal transmissions in the same channel.

Considering the agents' arm-pulling decisions and the arms' maximal capacities, the platform needs to optimize the payment setting and motivate agents to pull the corresponding arms. If the platform sets the payment on one arm too high, many agents may choose to pull that arm, which might negatively affect the total revenue of the platform. Therefore, the platform needs to judiciously design a payment policy to maximize the risk-aware revenue, considering the different risk-aware MAMAB scenarios.

1.2 Contributions

The key contributions are as follows:

- *Risk-aware Multi-agent multi-armed bandits:* To the best of our knowledge, this is the first work that considers risk-aware multi-agent multi-armed bandits, which includes arms with unknown capacities and game-theoretical agents with heterogeneous eligible arm sets. Our work bears important practical implications for the platform's optimal payment designs and opens up an exciting direction in practice.
- *Heterogeneous agents' arm-pulling decisions under Nash equilibrium:* Agents are heterogeneous in their qualification/capability of pulling arms, i.e., agent will have different eligible arm sets. In the presence of the coupling of agents' arm-pulling decisions, we show how to calculate their decisions under Nash equilibrium.
- *Payment algorithm with sub-linear regret under risk-aware MAMAB with independent risks:* Consider independent risks when multiple agents pull multiple arms at the same time step. We design an optimal payment algorithm to maximize the platform's risk-aware revenue (or minimize the regret), under both cases where the platform 1) can differentiate the utility on each arm, and 2) only observes the aggregate of the utility. We prove that both algorithms achieve the $O(\log T)$ regret.
- *Payment algorithm with sub-linear regret under risk-aware MAMAB with correlated risks:* We design an optimal payment algorithm to maximize the platform's risk-aware revenue (or minimize the regret), considering correlated risks when multiple agents pull multiple arms in the same time step. We prove that the algorithm achieves the $O(\sqrt{T}\log T)$ regret.

2 RELATED WORK

Our work focus on risk-aware MAMAB system. Here we group the related works from two perspectives: (1) MAMAB systems with game-theoretical agents; (2) risk-aware bandits.

The concept of MAMAB was first proposed by Liu and Zhao [6] and Gai *et al.* [7]. There are some MAMAB studies taking game-theoretical agents into consideration. For example, Boursier and Perchet [8] considered selfish agents who aim to maximize their individual payoffs. Tossou *et al.* [9] focused on a two-agent MAB problem and presented the bargaining solution of the game. Taywade *et al.* [10] investigated the modeling of Cournot games in MAMAB setting. Liu *et al.* [11, 12] and Sankararaman *et al.* [13] focused on matching markets, where both arms and agents have matching preferences when choosing each other. These papers only focused on expected reward maximization, but did not consider risk-aware MAMAB system.

Sani *et al.* [3] first introduced and formulated the MV risk-aware bandits problem. Vakili & Zhao [2, 14] and Liu *et al.* [4] studied this problem and completed the regret analysis. Zhu & Tan [15] assessed the possibility of Thompson sampling methods to solve this problem. In addition to MV, the conditional value at risk (CVaR) is also a useful measure to quantify the risk [16–18]. But these works did not consider the multi-agent setting or the correlated risks. Du *et al.* [19] allowed a learner to choose different arms at each time step and considered the arm correlation. Our work focuses on the more general setting under unknown arm capacities, and game-theoretical agents with different eligible arm sets.

3 SYSTEM MODEL

In this section, we provide an overview of the risk-aware MAMAB system. The system includes a platform associated with a set $\mathcal{K} = \{1, \ldots, K\}$ of K arms, and a set $\mathcal{N} = \{1, \cdots, N\}$ of N agents. We also consider a finite time horizon $\mathcal{T} = \{1, \ldots, T\}$. We first introduce the heterogeneous arms and agents on the platform in Sections 3.1 and 3.2, respectively. Then we present the platforms' risk-aware revenue maximization problem in Section 3.3.

3.1 Arms on the platform

There are a total of K arms on the platform waiting for N agents to select and pull. Each arm $k \in \mathcal{K}$ is associated with attributes

(m^k, X^k), where m^k denotes the maximal capacity on arm-selecting agents that this arm can support, while X^k is the "per-agent" stochastic utility on this arm which follows a Gaussian distribution. Denote the utility mean of each arm k as $\mu^k \triangleq \mathbb{E}[X^k]$. Utility mean μ^k and capacity m^k are unknown to both agents and the platform. These K arms are associated with a positive semi-definite covariance matrix Σ, where $\Sigma^{k,k} \leq 1$ for any $k \in \mathcal{K}$ without loss of generality.

Note that many prior papers ignore that agents are unwilling to pull the arms, as pulling arms usually incurs a certain cost (e.g., operational or battery cost). In this case, let us define the cost c^k associated with each arm $k \in \mathcal{K}$, which can model the cost of pulling this arm. Correspondingly, the platform will decide the payment $r_t^k \in \mathbb{R}$ on each arm k at time slot t, so to motivate agents in pulling each arm k.

3.2 Agents' arm-pulling decisions

In the multi-agent system, we consider heterogeneous agents such that only the VIP agents can access all arms, while non-VIP agents can only access subset of arms. Let us define the VIP arm set $\mathcal{K}^H$ that only agents in VIP set $\mathcal{N}^H$ can select from. And the number of VIPs is $N^H = |\mathcal{N}^H|$. The non-VIP arm set $\mathcal{K}\backslash\mathcal{K}^H$ can be accessed by all the agents (i.e., both VIPs and non-VIPs) in set $\mathcal{N}$.

Here we define the arm selection decision for each agent. Each VIP agent $n \in \mathcal{N}^H$ chooses an arm strategy $s_{n,t} \in \mathcal{S}_{n,t} \triangleq \mathcal{K} \bigcup \{0\}$ at time t. Here $s_{n,t} = 0$ means not pulling any arm, and $s_{n,t} = k \in \mathcal{K}$ means pulling arm k. Each non-VIP agent $n' \in \mathcal{N}\backslash\mathcal{N}^H$ can only access non-VIP arms, and chooses the action $s_{n',t} \in \mathcal{S}_{n',t} \triangleq \mathcal{K}\backslash\mathcal{K}^H \bigcup \{0\}$ at time t. Let $\boldsymbol{s}_{-n,t} = (s_{1,t}, \cdots, s_{n-1,t}, s_{n+1,t}, \cdots, s_{N,t})$ be the strategy profile of all other agents except agent n at time slot t. Note that an agent can only pull one arm at each time slot.[1] Based on whether agent n chooses arm k at time slot t, i.e., $\mathbf{1}\{a_{n,t} = k\}$, where $\mathbf{1}\{\cdot\}$ is the indicator function. The number of agents pulling arm k at time t is

$$n_t^k \triangleq \sum_{n \in \mathcal{N}} \mathbf{1}\{a_{n,t} = k\}. \tag{1}$$

When multiple agents select the same arm to pull, the platform will equally divide the payment among all the agents pulling this arm. Recall that the payment on arm k is r_t^k. Given the strategy profile $\boldsymbol{s}_{-n,t}$ and payment profile $\boldsymbol{r}_t \triangleq (r_t^k, \forall k \in \mathcal{K})$, the payoff of agents n pulling arm k at time slot t is:

$$\pi_{n,t}(s_{n,t}, \boldsymbol{s}_{-n,t}, \boldsymbol{r}_t) = \begin{cases} \frac{r_t^k}{n_t^k} - c^k, & \text{if } s_{n,t} = k, \\ 0, & \text{if } s_{n,t} = 0. \end{cases} \tag{2}$$

Choosing arm k yields a payoff from the difference between the allocated payment $\frac{r_t^k}{n_t^k}$ and the cost c^k; while not choosing any arm brings zero payoff. To maximize his payoff, an agent will decide his arm-pulling strategy, by both assessing the payments and anticipating other agents' strategies. Note that the payoff on arm k depends on the payment, cost, and number of agents pulling arm k. It is irrelevant to the capacity m^k on arm k.

[1] In this model, each agent can only pull one arm at each time slot. Nonetheless, we can extend this model to the setting that one agent can pull multiple arms at each time slot. This setting is equivalent to our model but with multiple agents. More specifically, an agent pulling arms 1, 2 and 3, is equivalent to the setting that 3 agents pull arms 1, 2, and 3 separately.

3.3 Platform's risk-aware problem

Now we focus on the platform's problem formulation, considering the agents' arm-pulling decisions, arms' maximal capacities, and the risk-aware MAB settings.

Let us first consider the utility on each arm. Given the capacity m^k and pulling number n_t^k on arm k, the effective number on arm k at time t is

$$n_t^{k,eff} \triangleq \min\{m^k, n_t^k\}, \tag{3}$$

where the effective number is bounded by the maximal capacity m^k. Then the platform obtains utility from arm k as

$$U_t^k \triangleq n_t^{k,eff} X_t^k. \tag{4}$$

This is motivated by many practical scenarios, i.e., an edge computing node can only serve a finite number of tasks. Recall the payment r_t^k on arm k, then the revenue on arm k is

$$V_t^k \triangleq U_t^k - r_t^k. \tag{5}$$

In risk-aware MABs with a single agent pulling one arm at each time step, an agent prefers to pull the arms with higher mean and lower uncertainty. To measure the risk, the mean-variance (MV) of an arm k captures a linear combination of the mean and the variance of the utility, which is defined as [4]

$$\eta^k = \rho\mu^k - \Sigma^{k,k}, \tag{6}$$

where $\rho \geq 0$ is a risk-tolerance factor. When $\rho \to \infty$, the risk-aware problem degenerates to a risk-neutral one; When $\rho = 0$, the problem aims to find the arm with the lowest risk.

Next, we consdier the following cases to discuss the risk-aware MAMAB systems respectively.

3.3.1 Independent risks: Cumulative revenue with MV of N agents. Here we consider the scenario where N agents pull multiple arms at the same time, and the risks of pulling multiple arms are independent. The platform can differentiate the generated utility on each arm. Then the platform calculates the cumulative revenue with MV generated from each agent, and aims to maximize the summation from all the agents. Following the payment policy $\boldsymbol{r}_t$ at time t, the revenue on arm k is $V_t^k(\boldsymbol{r}_t)$, and the number of pulling arm k is $n_t^k(\boldsymbol{r}_t)$. Each agent equally generates the revenue $\frac{V_t^k(\boldsymbol{r}_t)}{n_t^k(\boldsymbol{r}_t)}$ on his selected arm k at each time slot t.[2] And we define the revenue with MV at time t for agent n as $V_{n,t}(\boldsymbol{r}_t) \triangleq \sum_{k \in \mathcal{K}} \mathbf{1}\{a_{n,t} = k\} \cdot \frac{V_t^k(\boldsymbol{r}_t)}{n_t^k(\boldsymbol{r}_t)}$. Therefore, agent n's cumulative generated revenue with MV under policy $\boldsymbol{r}_t$ can be expressed as

$$\eta_{n,\boldsymbol{r}}(T) = \mathbb{E}\left[\sum_{t=1}^{T}\left(\rho V_{n,t}(\boldsymbol{r}_t) - \left(V_{n,t}(\boldsymbol{r}_t) - \frac{1}{T}\sum_{t=1}^{T} V_{n,t}(\boldsymbol{r}_t)\right)^2\right)\right]. \tag{7}$$

The platform aims to maximize the cumulative revenue with MV of all the agents under policy $\boldsymbol{r}_t$:

$$\eta_{\boldsymbol{r}}^{multi}(T) = \sum_{n \in \mathcal{N}} \eta_{n,\boldsymbol{r}}(T). \tag{8}$$

We also define a regret metric to quantify the performance of an algorithm when comparing with the optimal policy $\boldsymbol{r}^*$. Correspondingly, the optimal number vector is $\boldsymbol{n}^*(\boldsymbol{r}^*)$.

$$Reg_{\boldsymbol{r}}^{multi}(T) \triangleq \eta_{\boldsymbol{r}^*}^{multi}(T) - \eta_{\boldsymbol{r}}^{multi}(T). \tag{9}$$

[2] Without loss of generality, we consider the case that agents pulling the same arm equally generate the revenue on that arm, as agents are symmetric in pulling arms.

3.3.2 Independent risks: Cumulative revenue summation with MV. Here we consider the scenario where N agents collaboratively pull the arms, and the platform focuses on the revenue summation at each time slot. In this scenario, the platform can only observe the summation, but cannot differentiate the revenue of each arm. The platform aims to maximize the revenue summation with MV. Recall the revenue on arm k at time t as $V_t^k(\boldsymbol{r}_t)$ following the policy $\boldsymbol{r}_t$, hence the platform observes the revenue summation $V_t^{sum}(\boldsymbol{r}_t) \triangleq \sum_{k\in\mathcal{K}} V_t^k(\boldsymbol{r}_t)$ of all the arms. After pulling arms for T rounds, the cumulative revenue with MV under policy $\boldsymbol{r}_t$ is

$$\eta_{\boldsymbol{r}}^{sum}(T) = \mathbb{E}\left[\sum_{t=1}^{T}\left(\rho V_t^{sum}(\boldsymbol{r}_t) - \left(V_t^{sum}(\boldsymbol{r}_t) - \frac{1}{T}\sum_{t=1}^{T} V_t^{sum}(\boldsymbol{r}_t)\right)^2\right)\right]. \tag{10}$$

The platform aims to design a payment policy to maximize the cumulative revenue with MV. Here we define the optimal policy $\boldsymbol{r}^*$ as the benchmark. We also define a regret metric to quantify the performance of an algorithm when comparing with the optimal policy $\boldsymbol{r}^*$,

$$Reg_{\boldsymbol{r}}^{sum}(T) \triangleq \eta_{\boldsymbol{r}^*}^{sum}(T) - \eta_{\boldsymbol{r}}^{sum}(T). \tag{11}$$

3.3.3 Correlated risks: Cumulative revenue with MCV. Now we consider the case when N agents collaboratively pull arms and these arms can impact with each other via the covariance. Note that the platform can differentiate the utility generated on each arm. Define the mean vector $\boldsymbol{\mu} = (\mu^k, \forall k \in \mathcal{K})$ and the maximal capacity vector $\mathbf{m} = (m^k, \forall k \in \mathcal{K})$. The number vector under such policy $\boldsymbol{r}_t$ at time t is: $\mathbf{n}_t(\boldsymbol{r}_t) = (n_t^k(\boldsymbol{r}_t), \forall k \in \mathcal{K})$. Then we define the effective number vector at time t as $\mathbf{n}_t^{eff}(\boldsymbol{r}_t) = \min\{\mathbf{m}, \mathbf{n}_t(\boldsymbol{r}_t)\}$. Recall the covariance matrix Σ. For any policy $\boldsymbol{r}_t$, the instantaneous risk-aware revenue at time t with mean-covariance is [19]

$$f^{CV}(\boldsymbol{r}_t) = \left[\mathbf{n}_t^{eff}(\boldsymbol{r}_t)\right]^\top(\boldsymbol{\mu} - \boldsymbol{r}_t) - \rho\left[\mathbf{n}_t^{eff}(\boldsymbol{r}_t)\right]^\top \Sigma\left[\mathbf{n}_t^{eff}(\boldsymbol{r}_t)\right]. \tag{12}$$

The regret metric to quantify the performance of an algorithm when comparing with the optimal policy $\boldsymbol{r}^*$ is,

$$Reg_{\boldsymbol{r}}^{CV}(T) \triangleq \sum_{t=1}^{T}\mathbb{E}[f^{CV}(\boldsymbol{r}^*) - f^{CV}(\boldsymbol{r}_t)]. \tag{13}$$

Note that throughout this work, we consider the case with heterogeneous agents and arms with unknown maximal capacities. We first discuss the agents' arm-pulling decisions in Section 4. Then we discuss the platform's problem formulation in Section 5. And we analyze the platform's payment policy under three scenarios in Sections 6, 7, and 8, respectively.

4 AGENTS' ARM-PULLING DECISIONS

In this section, we focus on agents' arm-pulling decisions. We first formulate the agents' interactions as a non-cooperative game in Section 4.1 and analyze the agents' arm-pulling equilibrium strategy in Section 4.2.

4.1 Agents' arm pulling game

Agents make their arm selection decisions by participating in a non-cooperative game [20].

Before introducing the agents' arm-pulling decisions, we need to first define the best response strategy of each agent. Under a given strategy profile $\boldsymbol{s}_{-n,t}$, there are other $\hat{n}_t^k = |\{n' \in \mathcal{N}\backslash\{n\} : s_{n',t} = k\}|$ agents (except agent n) selecting arm k at time slot t. Given the payments $\boldsymbol{r}_t$ and strategy profile $\boldsymbol{s}_{-n,t}$, agent $n \in \mathcal{N}$ calculates his best response strategy to maximize payoff in (2). The fixed point of all the agents' best response choices is the Nash equilibrium (NE), where no agent can improve his payoff by deviating from his arm selecting choice unilaterally.

Definition 1 (Arm Pulling Nash equilibrium). *Given payments $\boldsymbol{r}_t$ at time slot t, a strategy profile $\boldsymbol{s}_t^{NE} = (s_{n,t}^{NE}, \forall n \in \mathcal{N})$ is an NE of game Ω_t if*

$$\pi_{n,t}(s_{n,t}^{NE}, \boldsymbol{s}_{-n,t}^{NE}, \boldsymbol{r}_t) \geq \pi_{n,t}(s_{n,t}, \boldsymbol{s}_{-n,t}^{NE}, \boldsymbol{r}_t), \quad \forall s_{n,t} \in \mathcal{S}_{n,t}, \forall n \in \mathcal{N}. \tag{14}$$

Given the agents' arm selection profile $\boldsymbol{s}_t^{NE}$ at time slot t, the number of agents selecting arm k under NE is

$$n_t^{k,NE} = |\{n \in \mathcal{N} : s_{n,t}^{NE} = k\}|. \tag{15}$$

4.2 Nash Equilibrium

In this section, we analyze the agents' arm pulling decisions under NE. The following Proposition characterizes the relationship between the platform's payments and the agents' NE arm selection decisions. Note that we first group the low-payment arm set as $\mathcal{K}_t^{low} \triangleq \{k \in \mathcal{K} : r_t^k < c^k\}$, where the payment is lower than the cost of pulling this arm k.

Proposition 1. *Given any payment profile $\boldsymbol{r}_t$, the number of agents selecting arm k under NE is $n_t^{k,NE}$ if and only if:*

i) for any arm $k \in \mathcal{K}_t^{low}$, we have $n_t^{k,NE} = 0$; and

ii) for any arm $k \in \mathcal{K}\backslash\mathcal{K}_t^{low}$, we have $n_t^{k,NE}$ such that

$$\sum_{k\in\mathcal{K}} n_t^{k,NE} \leq N, \sum_{k\in\mathcal{K}^H} n_t^{k,NE} \leq N^H, \tag{16}$$

$$\frac{r_t^k}{n_t^{k,NE}} - c^k = \lambda_t^k, \text{ where } \lambda_t^k \geq 0, \forall k \in \mathcal{K}\backslash\mathcal{K}_t^{low}, \tag{17}$$

$$\max_{k\in\mathcal{K}\backslash(\mathcal{K}_t^{low}\cup\mathcal{K}^H)} \frac{r_t^k}{n_t^{k,NE}+1} - c^k < \min_{k\in\mathcal{K}\backslash(\mathcal{K}_t^{low}\cup\mathcal{K}^H)} \lambda_t^k, \tag{18}$$

$$\max_{k\in\mathcal{K}\backslash\mathcal{K}_t^{low}} \frac{r_t^k}{n_t^{k,NE}+1} - c^k < \min_{k\in\mathcal{K}^H\backslash\mathcal{K}_t^{low}} \lambda_t^k. \tag{19}$$

Remark: All the proofs are given in [21]. Proposition 1(i) shows that no agents will pull arm k if the payment on that arm is too low. Constraint (16) denotes that each agent can pull at most one arm. The number of agents pulling VIP arms cannot exceed the number of VIPs. For a high-payment arm k in Proposition 1(ii), constraint (17) shows that agents will have a non-negative payoff λ_t^k if they select arm k. Constraints (18) and (19) show that changing to another arm can never increase the agent payoff, compared with the payoff λ_t^k of choosing arm k. Note that non-VIPs can only choose arms in set $\mathcal{K}\backslash\mathcal{K}^H$ (constraint (18)), while VIPs can choose any arm in set $\mathcal{K}$(constraint (19)).

Proposition 1 characterizes the relationship between the platform's payment profile and the agents' arm-pulling decisions under NE. In this case, we are able to formulate the platform's risk-aware revenue maximization problem as follows.

5 PLATFORM'S PROBLEM FORMULATION

Considering the agents' arm selection decisions at NE (where no agent has incentive to deviate), the platform aims to compute the optimal payments $\boldsymbol{r}_t$ to maximize the risk-aware revenue. For example, the platform will minimize the regret of the cumulative revenue with MV:

$$\text{minimize } Reg_{\boldsymbol{r}}^{multi}(T) \tag{20a}$$

$$\text{subject to } n_t^k = 0, \forall k \in \mathcal{K}_t^{low}, \forall t \in \mathcal{T}, \tag{20b}$$

$$(16) - (19), \forall t \in \mathcal{T}, \tag{20c}$$

$$\text{variables: } r_t^k \geq 0, n_t^k \geq 0, n_t^k \in \mathbb{N}, \lambda_t^k \geq 0, \forall k \in \mathcal{K}, \forall t \in \mathcal{T}. \tag{20d}$$

For different objectives $Reg_{\boldsymbol{r}}^{sum}(T)$ and $Reg_{\boldsymbol{r}}^{CV}(T)$, we can directly substitute the objective function in (20a). It should be noted that once the payment profile $\boldsymbol{r}_t$ is fixed, the numbers of agents $\boldsymbol{n}_t = (n_t^k, \forall k \in \mathcal{K})$ pulling arms are also determined under NE based on Proposition 1. Although problem (20) is a *mixed integer programming* problem [22] and the variables $\boldsymbol{r}_t$ and $\boldsymbol{n}_t$ are tightly coupling, we will show how to exploit the special structure of the payment and number of agents pulling arms to simplify the problem formulation.

Proposition 2. *The payment r_t^k and the number of agents n_t^k for any arm k at any time t is optimal for Problem (20) only if*

$$r_t^k = c^k n_t^k, \forall k \in \mathcal{K}, \tag{21}$$

$$0 \leq n_t^k \leq m^k, n_t^k \in \mathbb{N}, \forall k \in \mathcal{K}, \tag{22}$$

$$\sum_{k\in\mathcal{K}} n_t^k \leq N, \sum_{k\in\mathcal{K}^H} n_t^k \leq N^H. \tag{23}$$

Remark: Proposition 2 shows the necessary condition of the optimal payments. It means that the platform needs to set the payment to compensate the agents' cost of pulling arms without any additional payoff.

Then the platform's optimal payment problem is equivalent to finding the optimal number of agents n_t^k pulling each arm k at any time t. Thus, we can substitute $r_t^k = c^k n_t^k$ for any arm k at time t and reformulate Problem (20) as:

$$\text{minimize } Reg_{\boldsymbol{r}}^{multi}(T) \tag{24a}$$

$$\text{subject to } \sum_{k\in\mathcal{K}} n_t^k \leq N, \sum_{k\in\mathcal{K}^H} n_t^k \leq N^H, \forall t \in \mathcal{T}, \tag{24b}$$

$$\text{variables: } 0 \leq n_t^k \leq m^k, \forall k \in \mathcal{K}, \forall t \in \mathcal{T}. \tag{24c}$$

To calculate the optimal offline solution of the system under independent risks, we need to calculate the "per-agent" revenue with MV $\eta^k = \rho(\mu^k - c^k) - \Sigma^{k,k}$ for each arm k. Without loss of generality, let us assume that the arms have a descending order such that: $\eta^1 \geq \eta^2 \geq \ldots \geq \eta^K > 0$.[3] For the system under correlated risks, we need to consider the whole effect of the payment profile $\boldsymbol{r}$, while the optimal payments also satisfy the constraints in Proposition 2.

[3]Note that we assume $\eta^k > 0$ for any arm k. Otherwise, it is trivial to consider arm k, as recruiting any agent will not generate enough revenue on that arm.

6 INDEPENDENT RISKS: CUMULATIVE REVENUE WITH MV OF N AGENTS

In this section, we focus on the risk-aware MAMAB system with independent risks. The platform can differentiate the generated utility on each arm. Then the platform calculates the cumulative revenue with MV generated from each agent, and aims to maximize the aggregation from all the agents. To achieve this, we first introduce the UCB values and approximated optimal solutions in Sections 6.1 and 6.2. Then we propose an algorithm in Section 6.3, and prove the regret upper bound in Section 6.4.

6.1 Notations and calculations

6.1.1 Define UCB value B_t^k of each arm k at time t. Before introducing the payment algorithm, let us first define some notations. As the maximal capacity on each arm is unknown, let us define $m_l^k(t)$ and $m_u^k(t)$ as the updated lower and upper bounds of the capacity on each arm $k \in \mathcal{K}$ at time t, where the bounds satisfy $1 \leq m_l^k(t) \leq m^k \leq m_u^k(t) \leq N^H$.[4] Recall that U_t^k is the generated utility by pulling arm k (from n_t^k agents) at time t. Given the updated bounds and the number of agents pulling arm k, there are two determined cases: 1) Consider the indicating function $\mathbf{1}\{n_t^k \leq m_l^k(t)\} = 1$: The number of agents on arm k does not exceed the maximal capacity; and 2) Consider the indicating function $\mathbf{1}\{n_t^k \geq m_u^k(t)\} = 1$: The number of agents on arm k reaches the maximal capacity. Then we define the per-agent empirical mean $\bar{\mu}_t^k$ and the empirical variance s_t^k of arm k at time t as

$$\bar{\mu}_t^k = \frac{1}{\tau_t^k} \sum_{t'=1}^{t} \mathbf{1}\{1 \leq n_{t'}^k \leq m_l^k(t')\} \cdot \frac{U_{t'}^k}{n_{t'}^k}, \tag{25}$$

$$s_t^k = \frac{1}{\tau_t^k - 1} \sum_{t'=1}^{t} \mathbf{1}\{1 \leq n_{t'}^k \leq m_l^k(t')\} \cdot \left(\frac{U_{t'}^k}{n_{t'}^k} - \bar{\mu}_t^k\right)^2, \tag{26}$$

where $\tau_t^k \triangleq \sum_{t' \leq t} \mathbf{1}\{1 \leq n_{t'}^k \leq m_l^k(t')\}$ is the number of times until time t, when arm k is pulled and the number of agents does not exceed the capacity (i.e., $\mathbf{1}\{1 \leq n_{t'}^k \leq m_l^k(t')\} = 1$).

Then we define the UCB value of each arm k at time t as

$$B_t^k = \rho\left(\bar{\mu}_t^k - c^k + \sqrt{\frac{\log t}{\tau_t^k}}\right) - \frac{(\tau_t^k - 1)s_t^k}{\chi_{1-\alpha,\tau_t^k-1}}, \tag{27}$$

where $\chi_{1-\alpha,\tau_t^k-1}$ is the upper α percent of the chi-square distribution with $(\tau_t^k - 1)$ degrees of freedom. In (27), the first term is the UCB value of the mean from Hoeffding inequality, and the second term is from the characteristics of the chi-square distribution.

6.1.2 Evaluate the maximal capacity. For each chosen arm, the platform needs to determine the maximal capacity. Then the platform sets enough payment on that arm to motivate enough agents to together pull that arm, and update the lower and upper bounds of the maximal capacity. To achieve this, we need to define the empirical mean $\bar{v}_t^k$ of arm k under maximal capacity at time t as

$$\bar{v}_t^k = \frac{1}{\iota_t^k} \sum_{t'=1}^{t} \mathbf{1}\{n_{t'}^k \geq m_u^k(t')\} \cdot U_{t'}^k, \tag{28}$$

[4]Without loss of generality, we assume that each arm can at most accommodate all the VIP agents to together pull this arm.

where $\iota_t^k \triangleq \sum_{t' \le t} \mathbf{1}\{n_{t'}^k \ge m_u^k(t')\}$ is the number of times until time t when the number of agents pulling arm k reaches the maximal capacity (i.e., $\mathbf{1}\{n_{t'}^k \ge m_u^k(t')\} = 1$).

Then we are able to calculate the lower and upper bounds of the maximal capacity m^k at time t as:

$$m_l^k(t) = \max\left\{m_l^k(t-1), \left\lceil \frac{\hat{v}_t^k}{\hat{\mu}_t^k + \phi(\tau_t^k, \delta) + \phi(\iota_t^k, \delta)} \right\rceil\right\}, \quad (29)$$

$$m_u^k(t) = \min\left\{m_u^k(t-1), \left\lceil \frac{\hat{v}_t^k}{\hat{\mu}_t^k - \phi(\tau_t^k, \delta) - \phi(\iota_t^k, \delta)} \right\rceil\right\}, \quad (30)$$

where $\phi(x, \delta) \triangleq \sqrt{(1+\frac{1}{x})\frac{\log(2\sqrt{x+1}/\delta)}{2x}}$.

6.2 Approximated optimal solution $\tilde{\boldsymbol{r}}^*$

In traditional risk-neutral MAB system, to always pull the arm with the highest mean is the optimal offline solution. In risk-aware systems, however, to always pull the arm with the highest MV is not optimal. For example, consider one agent and two arms with Gaussian distribution with parameters $\mu^1 = 10$, $\mu^2 = 11$, $\Sigma^{1,1} = 1$, and $\Sigma^{2,2} = 2.1$. Let us further assume $\rho = 1$, $T = 2$, and identical cost of pulling arms $c^1 = c^2 = 1$. Calculating the MV, it is easy to show that $\eta^1 = 8$ and $\eta^2 = 7.9$.

- The platform chooses the single-arm payment policy $\tilde{\boldsymbol{r}}^* = (1, 0)$ that motivates the agent to always play arm 1, which yields a cumulative revenue with MV $\eta_{\tilde{\boldsymbol{r}}^*} = 8$.
- Then we consider a policy $\boldsymbol{r}_1 = (1, 0)$ and $\boldsymbol{r}_2 = \mathbf{1}\{X_1^1 < 10.5\} \cdot (1, 0) + \mathbf{1}\{X_1^1 \ge 10.5\} \cdot (0, 1)$. That is, at time 1 the platform sets payment on arm 1. If the generated utility at time 1 is lower than 10.5, then the platform still sets payment on arm 1 at time 2; otherwise, the platform changes to set payment on arm 2. This yields a cumulative revenue with MV $\eta_{\boldsymbol{r}} > 8.3$.

The above example shows that always motivating the agent to pull the arm with the highest MV is not optimal. Nonetheless, we regard this payment policy selecting the highest MV arm as an approximated optimal solution $\tilde{\boldsymbol{r}}^*$, which is a good proxy of the optimal solution $\boldsymbol{r}^*$.

The above example shows the approximated optimal solution under a single-agent scenario, now we focus on our model with multiple agents. Recall that the maximal capacity on each arm is unknown. And some agents (i.e., non-VIPs) can only access a subset of arms (i.e., non-VIP arms), while other agents (i.e., VIPs) can access all the arms (i.e., both VIP and non-VIP arms).

Here we define the approximated optimal solution under the general risk-aware multi-agent setting with heterogeneous agents and unknown arm capacities. Let us first discuss the approximated optimal number of agents pulling arms.

- *Non-VIP agents*: Let us first sort the non-VIP arms by η^k, and we define the lowest favored arm index as Φ^{non} such that $\sum_{k=1}^{\Phi^{non}} m^k \ge N - N^H$ and $\sum_{k=1}^{\Phi^{non}-1} m^k < N - N^H$. Then for the non-VIP agents, the optimal number of agents' pulling arms would be exactly m^1 agents choosing arm 1, m^2 agents choosing arm 2, and so on, until $m^{\Phi^{non}-1}$ agents choosing arm $\Phi^{non} - 1$, and $N - \sum_{k=1}^{\Phi^{non}-1} m^k$ agents choosing arm Φ^{non}.
- *VIP agents*: Note that there may exist capacity on the non-VIPs' least favored arm Φ^{non}, since the number of agents does not exceed the maximal capacity. Then the platform will motivate VIP agents to pull high MV arms that still have capacity to accommodate agents. In this case, the platform sort these arms by η^k. The idea is similar to the non-VIP agents, where VIP agents are first assigned to the arms with the highest MV, then assigned to the arms with the second highest MV, until no agent is left.

Given the approximated optimal number of agents pulling arms $\tilde{\boldsymbol{n}}^* = (\tilde{n}^k, \forall k \in \mathcal{K})$, the approximated optimal payment policy is straightforward that $\tilde{\boldsymbol{r}}^* = (c^k \tilde{n}^k, \forall k \in \mathcal{K})$ from Proposition 2. Here we define the approximated optimal arm set as $\tilde{\mathcal{K}}^* \triangleq \{k \in \mathcal{K} : \tilde{n}^k > 0\}$ that agents pull arms in set $\tilde{\mathcal{K}}^*$ and do not pull arms in set $\mathcal{K} \backslash \tilde{\mathcal{K}}^*$. Then we define the highest chosen arm index as $\Phi^{high} \triangleq \max_{k \in \tilde{\mathcal{K}}^*} k$.

6.3 Algorithm design

Now we are able to propose Algorithm 1 to maximize the cumulative revenue with MV of N agents' decisions under independent risks. The general idea is as follows.

- *Initialization:* The initial payment on each arm at any time is zero. The upper and lower bounds of the maximal capacity on each arm are initialized as N^H and 1, respectively (line 1).
- *UCB value calculation:* To begin with, the platform assigns payments on each arm one by one, and one VIP agent pulls the arm without deviation (lines 2 and 3). Then the platform calculates the UCB value B_t^k for each arm $k \in \mathcal{K}$ at time t (lines 4-6).
- *Non-VIP agents and arms:* Similar to the approximated optimal setting, the platform first focuses on the non-VIP agents and arms. Based on the UCB value and the lower bound of the capacity, the platform chooses the highest $\hat{\Phi}^{non}$ arms and forms the set $\mathcal{E}^{non}$ (line 8). This setting guarantees that the number of pulling each arm does not exceed the maximal capacity. For the least favored arm $\hat{\Phi}^{non}$, the number of non-VIPs pulling this arm is $n_t^{\hat{\Phi}^{non}}$ and the platform sets the corresponding payment (line 9). For the other chosen non-VIP arms, the number is m_l^k and the platform assigns corresponding payments (lines 10 and 11).
- *VIP agents and arms:* Here we define the arm capacity $\hat{m}^k$ for VIP agents (line 7). Note that there still exists capacity on the non-VIPs' least favored arm $\hat{\Phi}^{non}$, since the number does not exceed the lower bound. Hence at most $\hat{m}^{\hat{\Phi}^{non}}$ VIPs can pull this arm (line 13). Then VIPs pull arms from the whole set $\mathcal{K}$, except those arms already reaching the lower bound (i.e, $\mathcal{E}^{non} \backslash \hat{\Phi}^{non}$). The platform forms the set $\mathcal{E}^{VIP}$ for VIP agents to choose from (line 14) and further add payments to these chosen arms (lines 15-17). Note that the payment setting follows from Proposition 2 and satisfies the VIP and capacity constraints in (22) and (23).
- *Agents assignment and generated utilities:* Once the payment profile is fixed, the number of agents pulling each arm is determined from Proposition 1. As the platform cares about each agent's cumulative MV, the platform should encourage each agent to stick to the same arm. For example, the platform needs two agents to pull arms 1 and 2 at every time slot, then he assigns agent 1 (or 2, respectively) to pull arm 1 (or 2, respectively) all the time. For non-VIP assignment (line 12), the platform assigns the non-VIP agents one by one (according to their index) to the non-VIP arms

Algorithm 1: Independent risks: Cumulative revenue with MV of N agents' decisions

Input: Total number of agents N; total number of VIPs N^H; arm set $\mathcal{K}$; VIP arm set $\mathcal{K}^H$;

1 Initialization: Set $r_t^k = 0$ for any $k \in \mathcal{K}$ and $t \in \mathcal{T}$; $m_l^k = 1$ and $m_u^k = N^H$ for any $k \in \mathcal{K}$;
2 **for** $t = 1 : K$ **do**
3 The platform sets payment c^k on arm $k = t$ and one VIP agent pulls arm k ;
  Output: Generated utility U_t^k on arm k ;
4 **while** $t \le T$ **do**
5 **foreach** $k \in \mathcal{K}$ **do**
6 Calculate B_t^k in (27) ;
7 Set $\hat{m}^k = m_l^k, \forall k \in \mathcal{K}$;
8 Based on B_t^k, the platform chooses the highest $\hat{\Phi}^{non}$ arms from set $\mathcal{K}\backslash\mathcal{K}^H$ such that $\sum_{k=1}^{\hat{\Phi}^{non}} m_l^k \ge N - N^H$ and $\sum_{k=1}^{\hat{\Phi}^{non}-1} m_l^k < N - N^H$, and forms set $\mathcal{E}^{non}$;
9 The platform sets $n_t^k = N - N^H - \sum_{k=1}^{\hat{\Phi}^{non}-1} m_l^k$ and assigns payment $c^k n_t^k$ on arm $k = \hat{\Phi}^{non}$;
10 **foreach** $k \in \mathcal{E}^{non}\backslash\hat{\Phi}^{non}$ **do**
11 The platform sets $n_t^k = m_l^k$ and assigns payment $c^k n_t^k$ on arm k;
12 ▷ Non-VIP agents assignment ;
13 For arm $\hat{\Phi}^{non}$, set $\hat{m}^{\hat{\Phi}^{non}} = m_l^{\hat{\Phi}^{non}} - n_t^{\hat{\Phi}^{non}}$;
14 The platform chooses the highest $\hat{\Phi}^{VIP}$ arms from set $\mathcal{K}\backslash\mathcal{E}^{non} \bigcup \hat{\Phi}^{non}$ such that $\sum_{k=1}^{\hat{\Phi}^{VIP}} \hat{m}^k \ge N^H$ and $\sum_{k=1}^{\hat{\Phi}^{VIP}-1} \hat{m}^k < N^H$, and forms set $\mathcal{E}^{VIP}$;
15 The platform sets $n_t^k = N^H - \sum_{k=1}^{\hat{\Phi}^{VIP}-1} \hat{m}^k$ and assigns payment $c^k n_t^k$ on arm $k = \hat{\Phi}^{VIP}$;
16 **foreach** $k \in \mathcal{E}^{VIP}\backslash\hat{\Phi}^{VIP}$ **do**
17 The platform sets $n_t^k = m_l^k$ and assigns payment $c^k n_t^k$ on arm k ;
18 ▷ VIP agents assignment ;
  Output: Generated utility U_t^k for any arm $k \in \mathcal{K}$;
19 $t = t + 1$;
20 Define the eligible arm set $\mathcal{E} \triangleq \mathcal{E}^{non} \bigcup \mathcal{E}^{VIP}$;
21 **foreach** $k \in \mathcal{E}$ **do**
22 **if** $m_l^k \ne m_u^k$ **then**
23 The platform sets $n_t^k = m_u^k$ and assigns payment $c^k n_t^k$ on arm k ;
  Output: Generated utility U_t^k on arm k ;
24 Update m_l^k and m_u^k in (29) and (30) ;
25 $t = t + 1$;

sorted by the B_t^k value, until the number on arm k reaches the lower bound $\hat{m}^k$ and all the agents are assigned. The assignment of VIPs is similar (line 18). Given the agents' arm-pulling decisions under this assignment, the platform can observe the utility U_t^k on each arm k. This completes the agent assignment and utility generation process at time t (line 19).

- *Maximal capacity update:* For all the chosen arms (lines 20 and 21), if the maximal capacity on one arm is unknown (line 22), then the platform will set a corresponding payment and assign agents to that arm (line 23), to learn the maximal capacity (line 24). The learning process of each arm consumes a time slot (line 25). Note that the confidence interval width of the capacity is smaller than 1 (i.e., $m_u^k - m_l^k < 1$) with a determined capacity m^k, when the learning proceeds more than $\log T$ times.

Note that the homogeneous agent setting is a special case of our model when all agents are VIPs, i.e., $N^H = N$ and $\mathcal{K}^H = \mathcal{K}$ in Algorithms 1, 2 and 3. Then, the non-VIP assignment process, e.g., lines 8-13 of Algorithm 1, can be ignored as the non-VIP arm set is empty, and there is no need to consider the payment setting or the agent assignment on non-VIP arms. The known capacity case is a special case of our model when the upper and lower bounds of the capacity are the same, i.e., $m_l^k = m_u^k = m^k$ in Algorithms 1, 2, and 3. Then the arm capacity learning process, e.g., lines 21-25 of Algorithm 1, can be ignored as the capacity on each arm is known to the platform.

6.4 Regret upper bound

In the following Theorem, we state a sub-linear regret upper bound for Algorithm 1.

Theorem 1. *Given any fixed N, N^H, K, T, $\boldsymbol{\mu}$, $\boldsymbol{c}$, $\boldsymbol{m}$, and Σ, the regret of Algorithm 3 is upper bounded by*

$$Reg_{\boldsymbol{r}}^{multi} \le \sum_{k\in\mathcal{K}} \frac{49\omega^k (m^k)^2 \log T}{(\mu^k)^2} + \left(N \sum_{k>1} \frac{\Gamma_{k,1}^2}{\Delta_{k,1}} + N + 1\right) + 2 \sum_{k\in\mathcal{K}\backslash\tilde{\mathcal{K}}^*} m^k (\Delta_{k,1} + \Gamma_{k,1}^2) \left(\left(\frac{4\rho^2}{(\Delta_{k,\Phi^{high}})^2} + CN\right) \log T + 2N + 1\right), \tag{31}$$

where $\omega^k \triangleq 2f^{CV}(\boldsymbol{r}^*) + 2c^k N^H$, $\Delta_{k,x} = \eta^x - \eta^k$, and $\Gamma_{k,x} = (\mu^x - c^x) - (\mu^k - c^k)$.

Remark: Algorithm 1 solves the risk-aware MAMAB model with heterogeneous agents and unknown capacities. Despite the heterogeneous agents, we still consider the regret of pulling sub-optimal arms by considering the VIP and non-VIP agent assignment. The first term in (31) is the regret when the platform learns the maximal capacity on all arms, which follows the $\log T$ order given the capacity learning process. This can be eliminated under the known capacity case. The second term represents the learning regret of the approximated optimal solution. This fixed term shows that the high-MV arm selection is a good proxy. The third term calculates the regret when the platform assigns payments to sub-optimal arms. These sub-optimal arms can be pulled at most $\log T$ times.

7 INDEPENDENT RISKS: CUMULATIVE REVENUE SUMMATION WITH MV

Now we consider the case when the platform can only observe the utility summation and aims to maximize the cumulative revenue summation with MV. Note that the optimal payment profile and

Algorithm 2: Independent risks: Cumulative revenue summation with MV

Input: Total number of agents N; total number of VIPs N^H; arm set $\mathcal{K}$; VIP arm set $\mathcal{K}^H$;

1 Initialization: Set $r_t^k = 0$ for any $k \in \mathcal{K}$ and $t \in \mathcal{T}$; $m_l^k = 1$ and $m_u^k = N^H$ for any $k \in \mathcal{K}$; $t = 1$;
2 **while** $t \le T$ **do**
3 **foreach** $\hat{\boldsymbol{n}} \in \hat{\mathcal{N}}(\boldsymbol{m}_l)$ **do**
4 The platform calculate $B_t^{\hat{\boldsymbol{n}}}$;
5 Chooses the vector $\hat{\boldsymbol{n}}$ with the highest UCB value $B_t^{\hat{\boldsymbol{n}}}$;
6 **foreach** $k \in \mathcal{K}$ **do**
7 The platform sets payment $c^k \hat{n}^k$ on arm k and assigns $\hat{n}^k$ agents to pull arm k ;
  Output: Generated utility $U_t^{\hat{\boldsymbol{n}}}$ under vector $\hat{\boldsymbol{n}}$;
8 $t = t + 1$;
9 **foreach** $k \in \mathcal{K}$ *and* $\hat{n}^k > 0$ **do**
10 **if** $m_l^k \neq m_u^k$ **then**
11 The platform sets payment $c^k m_u^k$ on arm k ;
  Output: Generated utility U_t^k on arm k ;
12 $t = t + 1$;
13 The platform sets payment $c^k m_l^k$ on arm k ;
  Output: Generated utility U_t^k on arm k ;
14 $t = t + 1$;
15 Update m_l^k and m_u^k in (29) and (30) ;

the approximated optimal profile are the same as the profiles when the platform can differentiate the generated utility on each arm.

7.1 Notations and calculations

As the utility on each arm follows an independent Gaussian distribution, the summation of the utility still follows the Gaussian distribution. In this case, the platform can simply regard the summations of the utility (with different arms selected) as new Gaussian distributions. Let us define the vector $\hat{\boldsymbol{n}} = (\hat{n}^k, \forall k \in \mathcal{K})$ which includes $\hat{n}^k$ agents on each arm k. This vector $\hat{\boldsymbol{n}}$ can be regarded as a potential arm for the platform. Then we define the vector set, which contains all the vector $\hat{\boldsymbol{n}}$ such that the number on each arm does not exceed the maximal capacity and the number of agents satisfies the VIP number and total number constraints:

$$\hat{\mathcal{N}}(\boldsymbol{m}_l) = \Big\{\hat{\boldsymbol{n}} : 0 \le \hat{n}^k \le m_l^k \text{ for any } k \in \mathcal{K}, \sum_{k\in\mathcal{K}} \hat{n}^k \le N, \sum_{k\in\mathcal{K}^H} \hat{n}^k \le N^H\Big\}, \tag{32}$$

where $\boldsymbol{m}_l = (m_l^k, \forall k \in \mathcal{K})$.

Given the vector $\hat{\boldsymbol{n}}$, let us define the empirical mean $\bar{\mu}_t^{\hat{\boldsymbol{n}}}$ and the empirical variance $s_t^{\hat{\boldsymbol{n}}}$ of vector $\hat{\boldsymbol{n}}$ at time t as

$$\bar{\mu}_t^{\hat{\boldsymbol{n}}} = \frac{1}{\tau_t^{\hat{\boldsymbol{n}}}} \sum_{t'=1}^{t} \mathbf{1}\{\hat{\boldsymbol{n}}\} \cdot U_{t'}^{\hat{\boldsymbol{n}}}, s_t^{\hat{\boldsymbol{n}}} = \frac{1}{\tau_t^{\hat{\boldsymbol{n}}} - 1} \sum_{t'=1}^{t} \mathbf{1}\{\hat{\boldsymbol{n}}\} \cdot \left(U_t^{\hat{\boldsymbol{n}}} - \bar{\mu}_t^{\hat{\boldsymbol{n}}}\right)^2, \tag{33}$$

where $\mathbf{1}\{\hat{\boldsymbol{n}}\}$ denotes whether the number of agents pulling arms follow the vector $\hat{\boldsymbol{n}}$. And $\tau_t^{\hat{\boldsymbol{n}}} \triangleq \sum_{t' \le t} \mathbf{1}\{\hat{\boldsymbol{n}}\}$ calculates the number of times until time t, when the number of agents pulling arms follow the vector $\hat{\boldsymbol{n}}$. Then we define the UCB value of vector $\hat{\boldsymbol{n}}$ at time t as[5]

$$B_t^{\hat{\boldsymbol{n}}} = \rho\left(\bar{\mu}_t^{\hat{\boldsymbol{n}}} - \sum_{k\in\mathcal{K}} c^k \hat{n}^k + \sqrt{\frac{\log t}{\tau_t^{\hat{\boldsymbol{n}}}}}\right) - \frac{(\tau_t^{\hat{\boldsymbol{n}}} - 1)s_t^{\hat{\boldsymbol{n}}}}{\chi_{1-\alpha,\tau_t^{\hat{\boldsymbol{n}}}-1}}, \tag{34}$$

where $\chi_{1-\alpha,\tau_t^{\hat{\boldsymbol{n}}}-1}$ is the upper α percent of the chi-square distribution with $(\tau_t^{\hat{\boldsymbol{n}}} - 1)$ degrees of freedom.

7.2 Algorithm design

Now we are able to propose Algorithm 2 to maximize the cumulative revenue summation with MV under independent risks. The general idea is as follows.

- *Choose the vector with the highest UCB*: For each possible vector $\hat{\boldsymbol{n}} \in \hat{\mathcal{N}}(\boldsymbol{m}_l)$, the platform calculates its UCB value and chooses the one with the highest UCB value (lines 3-5). Given the chosen vector, the platform sets the corresponding payment and assigns the agents to pull these arms. Note that in this section, the platform only cares about the summation, so the platform can ignore the agent index. It is because two agents (with any agent index) pulling two selected arms generate the same summation.
- *Maximal capacity update*: For all the chosen arms (line 9), if the maximal capacity on one arm is unknown (line 10), then the platform will set a payment that attracts m_u^k (or m_l^k, respectively) agents (line 11 (or 13, respectively)) to learn the maximal capacity (line 15). Note that the learning process of each arm consumes two time slots (lines 12 and 14).

7.3 Regret upper bound

In the following Theorem, we state a sub-linear regret upper bound for Algorithm 2.

Theorem 2. *Given any fixed N, N^H, K, T, $\boldsymbol{\mu}$, $\boldsymbol{c}$, $\boldsymbol{m}$, and Σ, the regret of Algorithm 2 is upper bounded by*

$$Reg_{\boldsymbol{r}}^{sum} \le \sum_{k\in\mathcal{K}} \frac{49\omega^k (m^k)^2 \log T}{(\mu^k)^2} + \sum_{\hat{\boldsymbol{n}}\in\hat{\mathcal{N}}(\boldsymbol{m}),\hat{\boldsymbol{n}}\neq\boldsymbol{n}^*} \frac{\Gamma_{\hat{\boldsymbol{n}},\boldsymbol{n}^*}^2}{\Delta_{\hat{\boldsymbol{n}},\boldsymbol{n}^*}} + 1 + \sum_{\hat{\boldsymbol{n}}\in\hat{\mathcal{N}}(\boldsymbol{m}),\hat{\boldsymbol{n}}\neq\boldsymbol{n}^*} (\Delta_{\hat{\boldsymbol{n}},\boldsymbol{n}^*} + \Gamma_{\hat{\boldsymbol{n}},\boldsymbol{n}^*}^2)\left(\left(\frac{4\rho^2}{(\Delta_{\hat{\boldsymbol{n}},\boldsymbol{n}^*})^2} + C\right)\log T + 3\right). \tag{35}$$

Remark: The regret analysis is similar to Theorem 1, while the number of potential arms in Algorithm 2 is much higher (e.g., N^K arms in Algorithm 2 compared with N arms in Algorithm 1).

8 CORRELATED RISKS: CUMULATIVE REVENUE WITH MCV

In this section, let us consider the case when N agents collaboratively pull arms and these arms can impact with each other via the covariance. The platform can differentiate the utility on each arm and aims to maximize the cumulative revenue with MCV.

[5]Note that if the platform has not chosen a vector $\hat{\boldsymbol{n}}$ until time t, then the UCB value equals to infinity $B_t^{\hat{\boldsymbol{n}}} = \infty$.

8.1 Notations and calculations

In this section, the platform still observes the utility on each arm, hence the notation of $\bar{\mu}_t^k$ remains unchanged in (25), which is the "per-agent" utility mean on arm k at time t. Then we define the "per-agent" empirical covariance $\bar{\Sigma}_t^{i,j}$ on arms i and j at time t as

$$\bar{\Sigma}_t^{i,j} = \frac{1}{\tau_t^{i,j}} \sum_{t'=1}^{t} \mathbf{1}\{n_{t'}^i > 0, n_{t'}^j > 0\} \cdot \left(\frac{U_{t'}^i}{n_{t'}^i} - \bar{\mu}_t^i\right)\left(\frac{U_{t'}^j}{n_{t'}^j} - \bar{\mu}_t^j\right), \quad (36)$$

where $\tau_t^{i,j} \triangleq \sum_{t' \le t} \mathbf{1}\{n_{t'}^i > 0, n_{t'}^j > 0\}$ is the number of times until time t, when arm i and j are together pulled.

Before calculating the UCB value of the vector $\hat{\boldsymbol{n}}$, let us define the confidence regions. First, we define the confidence region of the covariance as

$$g_t^{i,j} \triangleq 16 \max\left\{\frac{3\ln t}{\tau_{t-1}^{i,j}}, \sqrt{\frac{3\ln t}{\tau_{t-1}^{i,j}}}\right\} + \sqrt{\frac{48\ln^2 t}{\tau_{t-1}^{i,j}\tau_{t-1}^{i}}} + \sqrt{\frac{36\ln^2 t}{\tau_{t-1}^{i,j}\tau_{t-1}^{j}}}, \quad (37)$$

which is the (i,j)-th index on the confidence region matrix $\boldsymbol{g}_t$ at time t. And we define the confidence region of the empirical mean of the number vector $\hat{\boldsymbol{n}}$ as

$$E_t^{\hat{\boldsymbol{n}}} = \sqrt{2\beta(\delta_t)\left(\hat{\boldsymbol{n}}^\top D_{t-1}^{-1}\left(\lambda\Lambda_{\bar{\Sigma}_t} D_{t-1} + \sum_{t'=1}^{t} \bar{\Sigma}_{t'}^{\hat{\boldsymbol{n}}_{t'}}\right) D_{t-1}^{-1}\hat{\boldsymbol{n}}\right)}, \quad (38)$$

where $\lambda > 0$ is the regularization parameter, $\beta(\delta_t) = \ln(1/\delta_t) + K\ln(\ln t) + K/2\ln(1+e/\lambda)$ and $\delta_t = 1/(t\ln^2 t)$. D_t is the diagonal matrix such that $D_t^{i,i} = \tau_t^i$. Λ_A is a diagonal matrix with the same diagonal with matrix A. Then we can calculate the UCB value of the number vector $\hat{\boldsymbol{n}}$ as

$$BCV_t^{\hat{\boldsymbol{n}}} = \hat{\boldsymbol{n}}^\top(\bar{\boldsymbol{\mu}}_t - \boldsymbol{c}) + E_t^{\hat{\boldsymbol{n}}} - \rho\hat{\boldsymbol{n}}^\top\left(\bar{\Sigma}_t - \boldsymbol{g}_t\right)\hat{\boldsymbol{n}}, \quad (39)$$

where $\boldsymbol{c} = (c^k, \forall k \in \mathcal{K})$ is the cost vector on all the arms. Note that the vector $\hat{\boldsymbol{n}}$ and the set $\hat{\mathcal{N}}(\boldsymbol{m}_l)$ are the same in Algorithm 2.

8.2 Algorithm design

Now we are able to propose Algorithm 3 to maximize the cumulative revenue with MCV under correlated risks. The idea is as follows.

- *Initialization*: At each time slot (of the first K^2 slots), the platform sets the payments on one pair of arms and observes the corresponding utility (lines 3-6).
- *Choose the vector with the highest UCB*: For each possible vector $\hat{\boldsymbol{n}} \in \hat{\mathcal{N}}(\boldsymbol{m}_l)$, the platform calculates its UCB value and chooses the one with the highest UCB value (lines 8-10). Given the chosen vector, the platform sets the corresponding payment and assigns the agents to pull these arms. Note that in this section, the platform can also ignore the agent index.
- *Maximal capacity update*: The process of learning the maximal capacity (lines 14-18) is similar to that in Algorithm 1.

8.3 Regret upper bound

In the following Theorem, we state a sub-linear regret upper bound for Algorithm 3.

Theorem 3. *Given any fixed N, N^H, K, T, $\boldsymbol{\mu}$, $\boldsymbol{c}$, $\boldsymbol{m}$, and Σ, the regret of Algorithm 3 is upper bounded by*

$$Reg_{\boldsymbol{r}}^{CV} \le O\left(\sqrt{L(\lambda)(K^2 + ||\Sigma||_+)KT\ln^2 T} + \rho K\sqrt{T}\ln T\right), \quad (40)$$

Algorithm 3: MAMAB with mean-covariance

Input: Total number of agents N; total number of VIPs N^H; arm set $\mathcal{K}$; VIP arm set $\mathcal{K}^H$;

1 Initialization: Set $r_t^k = 0$ for any $k \in \mathcal{K}$ and $t \in \mathcal{T}$; $m_l^k = 1$ and $m_u^k = N^H$ for any $k \in \mathcal{K}$;
2 **for** $t = 1 : K^2$ **do**
3 **if** $\lceil \frac{t}{K} \rceil == (t \bmod K) + 1$ **then**
4 The platform sets payment c^k and assigns one agent to pull arm $k = \lceil \frac{t}{K} \rceil$;
5 **else**
6 The platform sets payments c^k and $c^{k'}$ and assigns two agents to pull arm $k = (t \bmod K) + 1$ and arm $k' = \lceil \frac{t}{K} \rceil$, respectively ;
  Output: Generated utility U_t^k on each pulled arm $k \in \left\{\lceil \frac{t}{K} \rceil, (t \bmod K) + 1\right\}$;
7 **while** $t \le T$ **do**
8 **foreach** $\hat{\boldsymbol{n}} \in \hat{\mathcal{N}}(\boldsymbol{m}_l)$ **do**
9 The platform calculate $BCV_t^{\hat{\boldsymbol{n}}}$;
10 Chooses the vector $\hat{\boldsymbol{n}}$ with the highest value $BCV_t^{\hat{\boldsymbol{n}}}$;
11 **foreach** $k \in \mathcal{K}$ **do**
12 The platform sets payment $c^k\hat{n}^k$ on arm k and assigns $\hat{n}^k$ agents to pull arm k ;
  Output: Generated utility U_t^k for each arm $k \in \mathcal{K}$;
13 $t = t + 1$;
14 **foreach** $k \in \mathcal{K}$ *and* $\hat{n}^k > 0$ **do**
15 Lines 22-25 in Algorithm 1 ;

Table 1: Independent arms

Arm index	1	2	3	4	5	6
Per-agent mean	0.95	0.85	0.75	0.65	0.55	0.45
Per-agent variance	0.15	0.2	0.16	0.15	0.05	0.12
Per-agent cost	0.27	0.36	0.18	0.3	0.25	0.28
Maximal capacity	2	2	1	3	3	2

where $L(\lambda) = (\lambda+1)(\ln(1+\lambda^{-1})+1)$ *and* $||\Sigma||_+ = \sum_{i,j\in\mathcal{K}} \max\{\Sigma^{i,j}, 0\}$.

Remark: The regret is higher than that under the independent risks, as learning the correlated risks also consumes time.

9 PERFORMANCE EVALUATIONS

In this section, we first model the crowdsourcing platform Foap that includes six independent tasks (or arms). The "per-agent" random utility follows the Gaussian distribution. The utility mean, variance, cost, and maximal capacity on each arm are shown in Table 1. We set $\rho = 1$ and focus on 3 VIPs and 3 non-VIPs (6 agents in total), where non-VIPs can only access arm set $\{1, 4, 5\}$.

In Fig. 1(a)-(c), we focus on the case when the platform can differentiate the utility on each arm (i.e., Algorithm 1).

In Fig. 1(a), we show that the platform should set payments on arms 1, 2, 3, and 5 under Algorithm 1, hence the cumulative payment

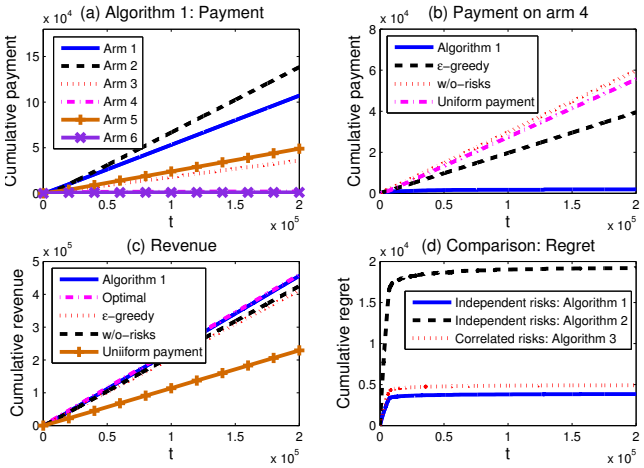

Figure 1: Regret vs. Time.

increases with time. Arms 4 and 6 do not generate enough revenue (difference between the utility, variance and payment), hence the platform will eventually not set any payment on these two arms, and the cumulative payment will no longer increase.

Then we compare Algorithm 1 with UCB algorithm without considering the risks (w/o-risks), ϵ-greedy and uniform payment algorithms. For ϵ-greedy algorithm (where we choose $\epsilon = 0.2$), the platform has $1 - \epsilon$ probability to assign payment on the highest revenue arms, and has ϵ probability to assign payment on other arms. For uniform payment algorithm, the platform has the same budget as our algorithm in each time slot, but distribute the payment on each arm uniformly.

We compare the payments on arm 4 in Fig. 1(b) and revenues in Fig. 1(c). Based on our algorithm, the platform eventually assigns zero payment on arm 4, while assigning payment to attract agents to pull arm 4 under w/o-risks, ϵ-greedy and uniform payment algorithms. To summarize, our algorithm guides the platform to set payments only on the arms with high risk-aware revenue, so as to attract agents to pull these arms, while w/o-risks algorithm focuses on the arms with high expected revenue, and ϵ-greedy and uniform payment algorithms cannot differentiate the optimal arms. In Fig. 1(c), the cumulative revenue of our algorithm approaches to that under the optimal setting, while w/o-risks, ϵ-greedy and uniform payment algorithms result in revenue losses.

Finally, we present the regret under Algorithms 1-3 in Fig. 1(d). For the correlated risk case, we consider an edge computing system with six correlated nodes (or arms). The covariance matrix Σ has the i, j-th entry equal to $0.005 \times i \times j$ where $i \neq j$. The other parameters are the same as in Table 1. We compare the two crowdsourcing scenarios with independent risks when the platform can or cannot differentiate the utility on each arm (i.e., Algorithm 1 or 2), and the edge computing scenario with correlated risks (i.e., Algorithm 3). We show that the three algorithms all achieve the sub-linear regret. The correlated risk scenario leads to a higher regret than the independent risk case, as learning the correlated risk also consumes time. The scenario when the platform cannot differentiate the utility results in the highest regret, as the number of potential arms is much higher than the other cases.

10 CONCLUSIONS

In this paper, we proposed a risk-aware multi-agent MAB (MAMAB) model, considering both the independent and correlated risk when multiple agents make arm-pulling decisions. We showed how to calculate the arm-pulling strategy of agents with potentially different eligible arm sets under a Nash equilibrium point. We designed the platform's optimal payment algorithms for its risk-aware revenue maximization (a regret minimization) under both independent and correlated risks. We proved that our algorithms achieve the sub-linear regret under independent risks when the platform can or cannot differentiate the utility on each arm. We also proved that our algorithm achieves the sub-linear regret under correlated risks.

REFERENCES

[1] A. Slivkins *et al.*, "Introduction to multi-armed bandits," *Foundations and Trends® in Machine Learning*, vol. 12, no. 1-2, pp. 1–286, 2019.
[2] S. Vakili and Q. Zhao, "Risk-averse multi-armed bandit problems under mean-variance measure," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 6, pp. 1093–1111, 2016.
[3] A. Sani, A. Lazaric, and R. Munos, "Risk-aversion in multi-armed bandits," *Advances in neural information processing systems*, vol. 25, 2012.
[4] X. Liu, M. Derakhshani, S. Lambotharan, and M. Van der Schaar, "Risk-aware multi-armed bandits with refined upper confidence bounds," *IEEE Signal Processing Letters*, vol. 28, pp. 269–273, 2020.
[5] Foap, https://www.foap.com/missions.
[6] K. Liu and Q. Zhao, "Distributed learning in multi-armed bandit with multiple players," *IEEE transactions on signal processing*, vol. 58, no. 11, pp. 5667–5681, 2010.
[7] Y. Gai, B. Krishnamachari, and R. Jain, "Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation," in *2010 IEEE Symposium on New Frontiers in Dynamic Spectrum (DySPAN)*. IEEE, 2010, pp. 1–9.
[8] E. Boursier and V. Perchet, "Selfish robustness and equilibria in multi-player bandits," in *Conference on Learning Theory*. PMLR, 2020, pp. 530–581.
[9] A. C. Tossou, C. Dimitrakakis, J. Rzepecki, and K. Hofmann, "A novel individually rational objective in multi-agent multi-armed bandits: Algorithms and regret bounds," in *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, 2020, pp. 1395–1403.
[10] K. Taywade, B. Harrison, and A. Bagh, "Modelling cournot games as multi-agent multi-armed bandits," *arXiv preprint arXiv:2201.01182*, 2022.
[11] L. T. Liu, H. Mania, and M. Jordan, "Competing bandits in matching markets," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 1618–1628.
[12] L. T. Liu, F. Ruan, H. Mania, and M. I. Jordan, "Bandit learning in decentralized matching markets," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 9612–9645, 2021.
[13] A. Sankararaman, S. Basu, and K. A. Sankararaman, "Dominate or delete: Decentralized competing bandits in serial dictatorship," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 1252–1260.
[14] S. Vakili and Q. Zhao, "Mean-variance and value at risk in multi-armed bandit problems," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2015, pp. 1330–1335.
[15] Q. Zhu and V. Tan, "Thompson sampling algorithms for mean-variance bandits," in *International Conference on Machine Learning*. PMLR, 2020, pp. 11 599–11 608.
[16] Y. David, B. Szörényi, M. Ghavamzadeh, S. Mannor, and N. Shimkin, "Pac bandits with risk constraints." in *ISAIM*, 2018.
[17] N. Galichet, M. Sebag, and O. Teytaud, "Exploration vs exploitation vs safety: Risk-aware multi-armed bandits," in *Asian conference on machine learning*. PMLR, 2013, pp. 245–260.
[18] A. Kagrecha, J. Nair, and K. P. Jagannathan, "Distribution oblivious, risk-aware algorithms for multi-armed bandits with unbounded rewards." in *NeurIPS*, 2019, pp. 11 269–11 278.
[19] Y. Du, S. Wang, Z. Fang, and L. Huang, "Continuous mean-covariance bandits," *Advances in Neural Information Processing Systems*, vol. 34, pp. 875–886, 2021.
[20] D. Fudenberg and J. Tirole, *Game Theory*. Cambridge, Massachusetts: MIT Press, 1991.
[21] Q. Shao, J. Ye, and J. C. Lui, Tech. Rep. [Online]. Available: https://www.dropbox.com/scl/fo/xuw3wmu2agtl3zmu2h6qg/AIA62vRHOVCDJWEm6UHF2-8?rlkey=nfhbva52li63pi8rvkp22a4gx&dl=0
[22] S. Burer and A. N. Letchford, "Non-convex mixed-integer nonlinear programming: A survey," *Surveys in Operations Research and Management Science*, vol. 17, no. 2, pp. 97–106, Jul. 2012.