Combinatorial Multi-armed Bandits in Competitive Environments

Submitted in partial fulfillment of the requirements for
the degreee of
Doctor of Philosophy
in
Electrical & Computer Engineering

Jinhang Zuo

B.E., Communication Engineering, Nanjing University of Posts and Telecom.
M.S., Electrical & Computer Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

September 2022

Abstract

Multi-armed bandits (MAB) have attracted much attention as a means of capturing the exploration and exploitation tradeoff in sequential decision making. In the classical MAB problem, at each round, a player chooses one arm from a fixed arm set and receives a random reward based on an unknown distribution. Nevertheless, in many real-world applications, the problems have a combinatorial nature among multiple arms and possibly non-linear reward functions. Combinatorial multi-armed bandits (CMAB) have been extensively studied for these settings, and most previous works consider CMAB from a single-player's perspective: at each round, one player chooses a set of arms to play, observes the feedback from them and receives a reward. However, motivated by applications such as online advertising (i.e., advertisers put ads on websites to attract user clicks), there might exist multiple players (advertisers) competing over the same set of arms (websites). This competition among players has been less studied and brings significant challenges to the design and analysis of bandit algorithms.

In this thesis, we introduce the competitive CMAB problem from two different perspectives. We first consider competitive CMAB from the follower's perspective, where a follower and a competitor play with the same set of arms. We assume the follower can choose his action after observing the action of the competitor and study how the follower can maximize his own reward given the competitor's actions. We then introduce competitive CMAB from the multi-players' perspective, where multiple players choose combinatorial actions on the same set of arms. Our objective is to design bandit algorithms that maximize the collective reward across all players. We provide general formulations of both settings and design bandit algorithms with theoretical guarantees for real-world applications, including social influence maximization, dynamic channel allocation, and general resource allocation.

Acknowledgments

First and foremost, I would like to thank my Ph.D. advisor, Prof. Carlee Joe-Wong, for her detailed guidance on my research and constant support throughout my Ph.D. journey. She is a great researcher as well as a kind advisor. When I started my Ph.D. study five years ago, I had no idea on how to do research. It was Carlee who taught me to think as a researcher and find good research questions. She was very patient in helping me solve technical problems. She always provided detailed comments on the papers we wrote and valuable feedback on my presentations. In my final year, she helped me a lot to find a position in academia and provided useful suggestions on my career plan. I am so fortunate to be advised by Carlee and cannot imagine a better advisor.

I am also grateful to my committee members, Osman Yağan, Guannan Qu, Wei Chen, and Xiaoxi Zhang, for their detailed comments and great suggestions on my research. It was their valuable feedback that brought this thesis to completion. My research was also partially supported by ONR Grant N000142112128, CNS Grant 2103024, and ARO Grant W911NF1910036.

I would like to thank my internship mentor, Wei Chen, for giving me the opportunity to work with him at Microsoft Research Asia in my second year. He is one of my researcher role models. I learned a lot from him not only about how to solve research problems but also about professional research attitude. Without his help, I could not set up my foundation in combinatorial bandits and influence maximization so quickly. We continued to collaborate after my internship and his suggestions always bring my research in the right direction. I also would like to thank my fellow group member Xutong Liu at MSRA. We keep collaborating on many research projects and have lots of discussions on studies, life, and future career plans.

I am fortunate to have Gauri Joshi, Osman Yağan, Samarth Gupta, Madhumitha Harishankar, Taejin Kim, Yuhang Yao, Yi Hu, Xiaoxi Zhang, John C.S. Lui, Shuai Li, Tong Yu, Songwen Hu, Handong Zhao, and Siwei

Wang as my collaborators. They are all brilliant, knowledgeable, and hardworking. We worked together on many exciting research projects, and I can always learn from their expertise in various research areas.

I will always miss the wonderful days in my Ph.D. study owing to the warm company of my fellow group members: Yichen Ruan, Taejin Kim, Yuhang Yao, Yi Hu, I-Cheng Lin, Madhumitha Harishankar, Tianshu Huang, Xiaoxi Zhang, Yuxuan Jiang, Marie T. Siew. Special thanks to Xiaoxi and Yuxuan for their valuable guidance on research during my first year at CMU. Thank Jingxian Wang, Jianyu Wang, Zinan Lin, Yuezhang Zou, Zifan Wang, Yuting Bu, Zhipeng Bao, and Shuo Li for their company in Pittsburgh. Thank Pai Zhu, Zekun Fan, Jianfeng Zhang, Xiaoyi Duan, Yuechen Luo, and Shi Shu for their company in Silicon Valley. I also would like to thank my old friends for the beautiful memories we shared: Jiawei Li, Rui Ge, Zixuan Zhou, Ruizhi Pu, Xiaowen Xiong, Jingchun Ma, Yutong Wang, Kang Yang, Di Wu, Tong Chen.

Lastly, I would like to thank my family for their love and support. I am grateful to my beloved parents who raise me up and always support my decisions. Without their understanding and support, I would never be able to finish this long Ph.D. journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The multi-armed bandit (MAB) problem has been extensively studied in statistics and machine learning. In the classical MAB problem, there are K arms, each having an unknown reward distribution. At each round, a player chooses one of these arms and receives a random reward drawn from its reward distribution. The goal is to maximize the long-term cumulative reward of the player over multiple rounds. The MAB problem captures the fundamental tradeoff between exploration and exploitation in sequential decision making: exploring unknown arms can potentially discover an arm with a higher reward while exploiting the best-known arm may avoid choosing arms with low rewards. Most MAB algorithms use the history of rewards received from the played arms to design strategies for choosing arms in future rounds. The performance of a bandit algorithm is measured by its expected *regret*, which is the difference in the expected cumulative reward between always playing the best arm and playing arms according to the algorithm. Existing results [1] show that one can achieve a T-round regret of $O(\log T)$, which is asymptotically optimal.

Nevertheless, in many real-world applications, the problems have a combinatorial nature among multiple arms and possibly non-linear reward functions. The combinatorial multi-armed bandit (CMAB) framework has been proposed for these settings [2, 3, 4] and has been applied to various applications in recommender systems [5], wireless networking [6], social networks [7], etc. In traditional CMAB problems, one player selects a combinatorial action (i.e., a combination of arms) to play in each

round, which would trigger a set of arms. The outcomes of the triggered arms are observed as feedback to the player. The player then uses the observed feedback to update his knowledge of the arms, which is used for action selection in later rounds.

Let us consider an example of traditional CMAB in online advertising. As shown in Figure 1.1, an advertiser wants to put advertisements on a set of m web pages to attract user clicks. Due to a budget constraint, the advertiser can choose at most k web pages. Each user has a click-through probability for the advertisement on the certain page he visited, but this probability is unknown by the advertiser. The users will visit these web pages repeatedly (i.e., homepages of news websites), and the goal of the advertiser is to maximize the total number of user clicks over multiple rounds. The advertiser needs to repeatedly select k web pages, observe the click results to learn the click-through probabilities, and decide which pages to choose in future rounds. Besides the exploration and exploitation tradeoff, CMAB has to deal with the exponential number of possible actions (e.g., $\binom{m}{k}$ choices of web pages in online advertising) that makes exploring all actions infeasible especially when m is extremely large.

Figure 1.1: Online advertising.

Another application of CMAB is in music recommendation. As shown in Figure 1.2, the recommender agent recommends an ordered list of k songs among m songs to the user. At each round, a user arrives and checks the recommended list from the first to the last. The user has a click probability on each song and will

stop checking the list once clicking a song. The agent observes the song on which the user clicks, and knows the user did not click on previous songs in the list. It uses this information to update its estimates of each song's click probabilities. The goal of the agent is to maximize the number of clicks over time without knowing the click probabilities. This problem is also called cascading bandits [5], which is a special case of CMAB. Same as the online advertising example, the possible number of actions is exponential (i.e., $\frac{m!}{(m-k)!}$). CMAB algorithms avoid the direct exploration of these actions via learning the click probabilities of individual songs and making decisions based on the learned statistics.

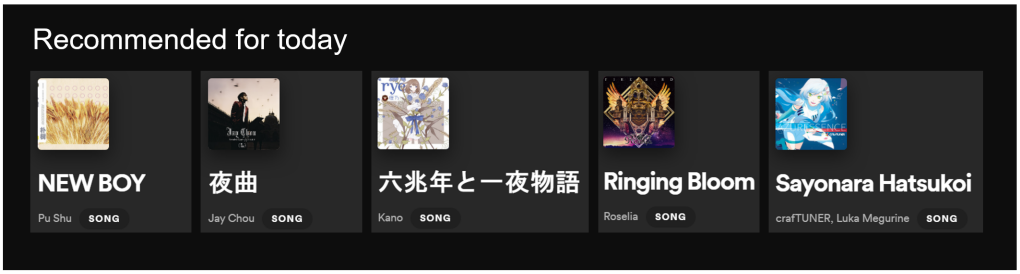


Figure 1.2: Music recommendation.

Most previous works consider CMAB problems from a single-player's perspective: in the example of online advertising, they only consider one advertiser putting a single type of advertisement on web pages. However, in the real-world advertising problem, there might exist different types of competing advertisements on these web pages, either provided by the same advertiser or other competing advertisers. This competition among players has been less studied and brings significant challenges to the design and analysis of CMAB algorithms.

In this thesis, we introduce two new competitive CMAB settings that explicitly model the competition between players.

- Competitive CMAB from the follower's perspective: in this setting, we consider a player and a competitor (or a group of competitors) playing with the same set of arms. Playing on the same arm incurs competition, which might lead to a potential loss of the reward. We call this competitive CMAB from the follower's perspective as we assume the player can choose her action after observing the action of the competitor and our objective is to maximize the collected reward of the player. In the online advertising example shown in

Figure 1.3, it can model that the advertiser puts her advertisement on web pages that may already have competing advertisements.

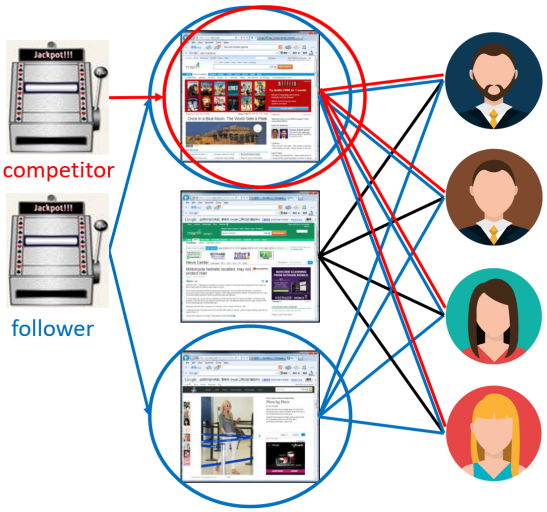


Figure 1.3: Online advertising from the follower's perspective.

- Competitive CMAB from the multi-players' perspective: in this setting, instead of making decisions for one single player, we need to choose combinatorial actions for multiple players who play with the same set of arms. Again, playing on the same arm incurs competition, which might lead to a potential loss of the reward. Our objective is to maximize the overall reward for all players. We consider both the centralized and the distributed settings. In the centralized settings, we assume there exists a central controller making decisions for all players and also observing the feedback from all players. In the online advertising example shown in Figure 1.4, it can model that the advertiser needs to put competing advertisements for different products on the web pages. In the distributed settings, each player chooses her action individually only based on her own feedback. Our goal is to find a learning policy that can be deployed on all players to maximize the overall reward. In the online advertising example, it can model that a group of advertisers put competing advertisements on the same set of web pages without any communication between each other.

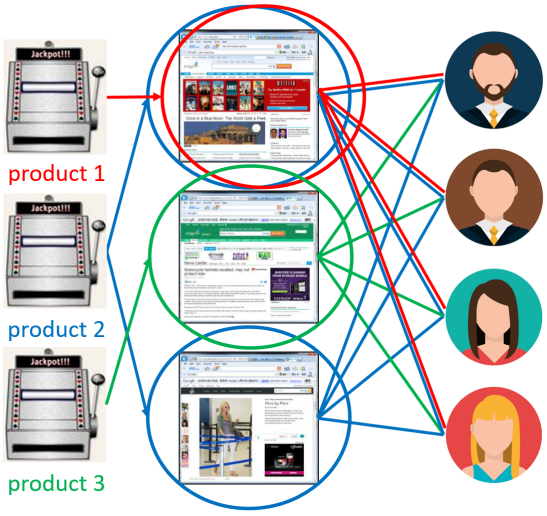


Figure 1.4: Online advertising from the multi-players' perspective.

Table 1.1: Summary of different settings.

Setting	Competitive	Combinatorial	Player	Decision Maker	Objective
Traditional CMAB [2, 3, 4]	✗	✓	single	single	single
Multi-player MAB [6, 8]	✓	✗	multiple	individual players	all players
Follower	✓	✓	follower & competitor	follower	follower
Multi-player: centralized	✓	✓	multiple	central controller	all players
Multi-player: distributed	✓	✓	multiple	individual players	all players

1.2 Overview of Our Approaches

In this section, we give an overview of our proposed approaches for different competitive CMAB problems. We first compare our proposed competitive settings with traditional CMAB and multi-player MAB settings in Table 1.1. We are the first to consider the competitive CMAB problem where both the follower and the competitor can take combinatorial actions. Different from previous multi-player bandits [6, 8], our competitive CMAB from the multi-players' perspective allow the players to take combinatorial actions in both centralized and distributed settings.

For the competitive CMAB from the follower's perspective, we study it with the application in social networks.

- **Online competitive influence maximization** [9]. Online influence maximization has attracted much attention as a way to maximize influence spread through a social network while learning the values of unknown network parameters. Most previous works focus on single-item diffusion. In this thesis, we

introduce a new Online Competitive Influence Maximization (OCIM) problem, where two competing items (e.g., products, news stories) propagate in the same network (as in Figure 1.5) and influence probabilities on edges are unknown. We consider one of the items as the follower. At each round, given the seed nodes of the competing item, the follower chooses k nodes as the seed set, observes the full diffusion results of both items, and uses them to learn the influence probabilities for future seed selection. The objective is to maximize the total number of nodes influenced by the follower over multiple rounds. Compared to the online advertising problem with a bipartite graph discussed above, the OCIM problem is more challenging as it considers the influence propagation of competing items in a general social graph. We adopt a combinatorial multi-armed bandit (CMAB) framework for OCIM, but unlike the non-competitive setting, the important monotonicity property (influence spread increases when influence probabilities on edges increase) no longer holds due to the competitive nature of propagation, which brings a significant new challenge to the problem. We provide a nontrivial proof showing that the Triggering Probability Modulated (TPM) condition for CMAB still holds in OCIM, which is instrumental for our proposed algorithms OCIM-TS and OCIM-OFU to provably achieve sublinear Bayesian and frequentist regret, respectively. We also design an OCIM-ETC algorithm that requires less feedback and easier offline computation, at the expense of a worse frequentist regret bound. Experimental evaluations demonstrate the effectiveness of our algorithms.

For competitive CMAB from the multi-players' perspective, we study it with the applications in dynamic channel allocation and general resource allocation.

- **Dynamic Channel Allocation with Pre-observations** [10]. We consider the stochastic multi-armed bandit (MAB) problem in a setting where a player can pay to pre-observe arm rewards before playing an arm in each round. The single-player version of this problem can also be viewed as cascading bandits with position discounts [11], while we extend it to the multi-player settings. The formulation is inspired by Cognitive Radio Networks (CRNs), where users can use wireless channels when they are unoccupied by primary users. In each round, a user can sense (pre-observe) some channels (arms) to check their availability (reward) before choosing a channel to transmit data (play). Sensing

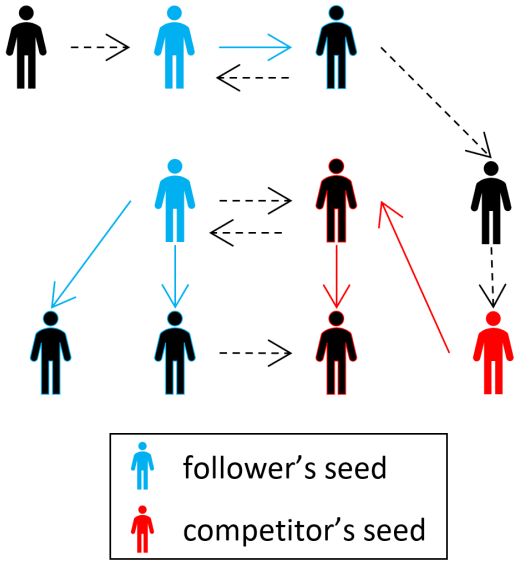


Figure 1.5: Example of online competitive influence maximization.

more arms leaves less time for data transmission, inducing a cost of making pre-observations. Figure 1.6 shows an example of such pre-observations. Apart from the usual trade-off between exploration and exploitation, we encounter an additional dilemma: pre-observing more arms gives a higher chance to play the best one, but incurs a larger cost. For the single-player setting, we design an OBP-UCB algorithm and prove a T-round regret upper bound $O(K^2 \log T)$. In the multi-player setting, collisions will occur when players select the same arm to play in the same round. We design a centralized algorithm, C-MP-OBP, and prove its T-round regret relative to an offline greedy strategy is upper bounded in $O(\frac{K^4}{M^2} \log T)$ for K arms and M players. We also propose distributed versions of the C-MP-OBP policy, called D-MP-OBP and D-MP-Adapt-OBP, achieving logarithmic regret with respect to collision-free target policies. Experiments on synthetic and real data show that C-MP-OBP and D-MP-OBP outperform random heuristics and offline optimal policies that do not allow pre-observations.

- **Resource Allocation** [12, 13]. We study the sequential resource allocation problem where a decision maker repeatedly allocates budgets between resources. Motivating examples include allocating limited computing time or wireless spectrum bands to multiple users (i.e., resources). As shown in Figure 1.7, at

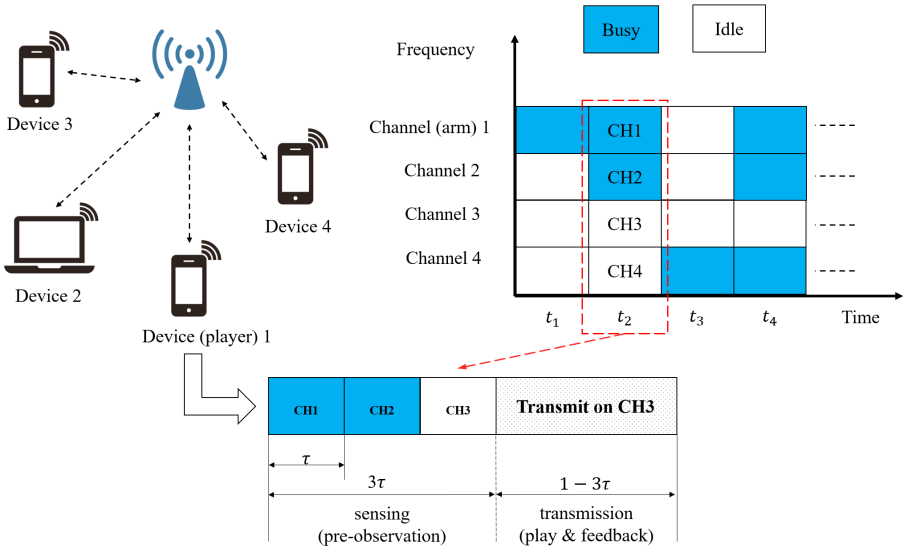


Figure 1.6: Illustration of pre-observations in a wireless network.

each timestep, the decision maker should distribute its available budgets among different resources to maximize the expected reward over time, or equivalently to minimize the cumulative regret. In doing so, the decision maker should learn the value of the budget allocated for each user from feedback on each user's received reward. For example, users may send messages of different urgency over wireless spectrum bands; the reward generated by allocating spectrum to a user then depends on the message's urgency. We assume each user's reward follows a random process that is initially unknown. We design combinatorial multi-armed bandit algorithms to solve this problem with discrete or continuous budget allocations. We prove the proposed algorithms achieve logarithmic regrets. In addition, since rewards received by the same user under different budget allocations are often correlated in practical settings, we propose a novel correlated combinatorial bandit algorithm that can achieve reduced regrets relative to correlation-agnostic algorithms. We demonstrate the effectiveness of all proposed algorithms through experiments for several wireless applications.

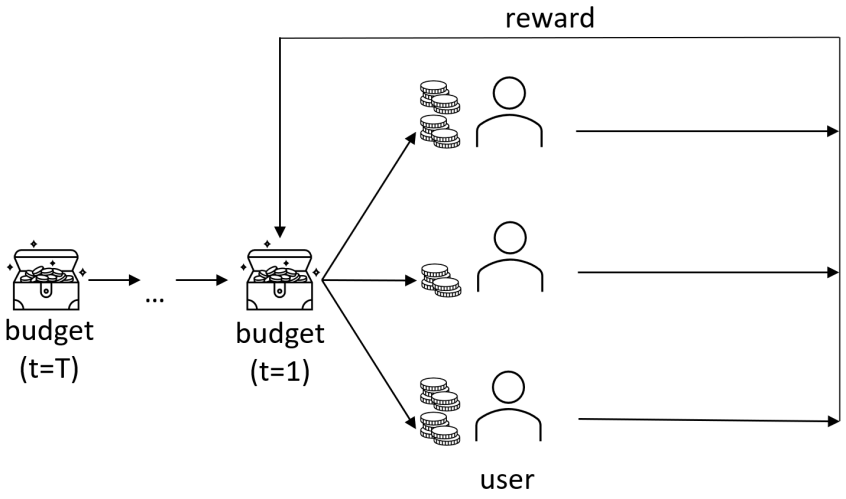


Figure 1.7: Sequential budget allocation.

1.3 Structure of the Thesis

The remainder of the thesis is organized as follows. Chapter 2 reviews the traditional non-competitive CMAB problem and introduces a new algorithm with an improved regret bound. Chapter 3 introduces the general formulation and algorithms for competitive CMAB from the follower's perspective and discusses its application in influence maximization. Chapter 4 introduces the general formulation for competitive CMAB from the multi-players' perspective and discusses its applications in wireless networking and general resource allocation. Chapter 5 concludes the thesis and discusses future research directions.

Chapter 2

Non-competitive CMAB

2.1 Introduction

In this section, we introduce the traditional non-competitive CMAB problem: in each round, one player selects a combinatorial action to play, which would trigger a set of arms. The outcomes of the triggered arms are observed as feedback to the player. The player then uses the observed feedback to update his knowledge of the arms, which is used for arm selection in later rounds. We first review the general CMAB problem formulation, then discuss the classical CUCB algorithm [2] and its regret bound. We also introduce a new TPVM bounded smoothness condition that helps reduce the regret dependency on the batch size and design a BCUCB-T algorithm with an improved regret bound [14]. Notice that such regret bound improvement may also be extended to the competitive CMAB problems in later chapters, but it requires a thorough discussion on the TPVM condition for these competitive settings, which is an interesting future direction.

2.2 Problem Formulation

We study the non-competitive combinatorial multi-armed bandit problem with probabilistic triggering arms (CMAB-T). Following the setting from [3], a CMAB-T problem instance can be described by a tuple $([m], \mathcal{S}, \mathcal{D}, D_{\text{trig}}, R)$, where $[m] = \{1, 2, ..., m\}$ is the set of base arms; $\mathcal{S}$ is the set of actions; $\mathcal{D}$ is the set of possible distributions over the outcomes of base arms with bounded support $[0, 1]^m$; D_{trig} is

the probabilistic triggering function and R is the reward function, the definitions of which will be introduced shortly.

In CMAB-T, the learning agent interacts with the unknown environment in a sequential manner as follows. First, the environment chooses a distribution $D \in \mathcal{D}$ unknown to the agent. Then, at round $t = 1, 2, ..., T$, the agent selects an action $S_t \in \mathcal{S}$ and the environment draws from the unknown distribution D a random outcome $\boldsymbol{X}_t = (X_{t,1}, ...X_{t,m}) \in [0,1]^m$. Note that the outcome $\boldsymbol{X}_t$ is assumed to be independent from outcomes generated in previous rounds, but outcomes $X_{t,i}$ and $X_{t,j}$ in the same round could be correlated. Let $D_{\text{trig}}(S, \boldsymbol{X})$ be a distribution over all possible subsets of $[m]$, i.e. its support is $2^{[m]}$. When the action S_t is played on the outcome $\boldsymbol{X}_t$, base arms in a random set $\tau_t \sim D_{\text{trig}}(S_t, \boldsymbol{X}_t)$ are triggered, meaning that the outcomes of arms in τ_t, i.e., $\{X_{t,i}\}_{i \in \tau_t}$ are revealed as the feedback to the agent, and are involved in determining the reward of action S_t. Function D_{trig} is referred as the probabilistic triggering function. At the end of the round t, the agent will receive a non-negative reward $R(S_t, \boldsymbol{X}_t, \tau_t)$, determined by $S_t, \boldsymbol{X}_t$ and τ_t. CMAB-T significantly enhances the modeling power of CMAB [15] and can model many applications such as cascading bandits and online influence maximization [3].

The goal of CMAB-T is to accumulate as much reward as possible over T rounds, by learning distribution D or its parameters. Let $\boldsymbol{\mu} = (\mu_1, ..., \mu_m)$ denote the mean vector of base arms' outcomes. Following [3], we assume that the expected reward $\mathbb{E}[R(S, \boldsymbol{X}, \tau)]$ is a function of the unknown mean vector $\boldsymbol{\mu}$, where the expectation is taken over the randomness of $\boldsymbol{X} \sim D$ and $\tau \sim D_{\text{trig}}(S, \boldsymbol{X})$. In this context, we denote $r(S; \boldsymbol{\mu}) \triangleq \mathbb{E}[R(S, \boldsymbol{X}, \tau)]$ and it suffices to learn the unknown mean vector instead of the joint distribution D, based on the past observation.

The performance of an online learning algorithm A is measured by its *regret*, defined as the difference of the expected cumulative reward between always playing the best action $S^* \triangleq \text{argmax}_{S \in \mathcal{S}} r(S; \boldsymbol{\mu})$ and playing actions chosen by algorithm A. For many reward functions, it is NP-hard to compute the exact S^* even when $\boldsymbol{\mu}$ is known, so similar to [3], we assume that the algorithm A has access to an offline (α, β)-approximation oracle, which for mean vector $\boldsymbol{\mu}$ outputs an action S such that $\Pr[r(S; \boldsymbol{\mu}) \geq \alpha \cdot r(S^*; \boldsymbol{\mu})] \geq \beta$. Formally, the T-round (α, β)-approximate regret is defined as

$$Reg(T; \alpha, \beta, \boldsymbol{\mu}) = T \cdot \alpha\beta \cdot r(S^*; \boldsymbol{\mu}) - \mathbb{E}\left[\sum_{t=1}^{T} r(S_t; \boldsymbol{\mu})\right], \tag{2.1}$$

where the expectation is taken over the randomness of outcomes $\boldsymbol{X}_1, ..., \boldsymbol{X}_T$, the triggered sets $\tau_1, ..., \tau_T$, as well as the randomness of algorithm A itself.

In the CMAB-T model, there are several quantities that are crucial to the subsequent study. We define triggering probability $p_i^{D,D_{\text{trig}},S}$ as the probability that base arm i is triggered when the action is S, the outcome distribution is D, and the probabilistic triggering function is D_{trig}. Since D_{trig} is always fixed in a given application context, we ignore it in the notation for simplicity, and use $p_i^{D,S}$ henceforth. Triggering probabilities $p_i^{D,S}$'s are crucial for the triggering probability modulated bounded smoothness conditions to be defined below.

Owing to the nonlinearity and the combinatorial structure of the reward, it is essential to give some conditions for the reward function in order to achieve any meaningful regret bounds [2, 3, 15]. The following are two standard conditions originally proposed by [3].

Condition 2.1 (Monotonicity)**.** *We say that a CMAB-T problem instance satisfies monotonicity condition, if for any action $S \in \mathcal{S}$, any two distributions $D, D' \in \mathcal{D}$ with mean vectors $\boldsymbol{\mu}, \boldsymbol{\mu}' \in [0,1]^m$ such that $\mu_i \leq \mu_i'$ for all $i \in [m]$, we have $r(S; \boldsymbol{\mu}) \leq r(S; \boldsymbol{\mu}')$.*

Condition 2.2 (1-norm TPM Bounded Smoothness)**.** *We say that a CMAB-T problem instance satisfies the triggering probability modulated (TPM) B-bounded smoothness condition, if for any action $S \in \mathcal{S}$, any distribution $D, D' \in \mathcal{D}$ with mean vectors $\boldsymbol{\mu}, \boldsymbol{\mu}' \in [0,1]^m$, we have $|r(S; \boldsymbol{\mu}') - r(S; \boldsymbol{\mu})| \leq B \sum_{i \in [m]} p_i^{D,S} |\mu_i - \mu_i'|$.*

The first monotonicity condition indicates the reward is larger if the parameter vector $\boldsymbol{\mu}$ is larger. The second condition bounds the reward difference caused by the parameter change (from $\boldsymbol{\mu}$ to $\boldsymbol{\mu}'$). One key feature is that the parameter change in each base arm $i \in [m]$ is modulated by the triggering probability $p_i^{D,S}$. Intuitively, for base arm i that is unlikely to be triggered/observed (small $p_i^{D,S}$), Condition 2.2 ensures that a large change in μ_i only causes a small change (multiplied by $p_i^{D,S}$) in the reward, and thus one does not need to pay extra cost to observe such arms. Many applications satisfy Condition 2.1 and Condition 2.2, such as linear combinatorial bandits [16], combinatorial cascading bandits [17], online influence maximization [3].

2.3 CUCB Algorithm and Regret Bound

In this section, we introduce the CUCB algorithm (Algorithm 2.1) in [2] for the CMAB-T problem. The algorithm maintains the empirical estimate $\hat{\mu}_i$ for the true mean μ_i, and feed the upper confidence bound $\bar{\mu}_i$ to the offline oracle to obtain the next action S to play. The upper confidence bound $\bar{\mu}_i$ is large if arm i is not triggered often (T_i is small), providing optimistic estimates for less observed arms. We next restate its improved regret bound in [3] owing to the 1-norm TPM bounded smoothness condition.

Definition 2.1 (Gap)**.** *Fix a distribution D and its expectation vector $\boldsymbol{\mu}$. For each action S, we define the gap $\Delta_S = \max(0, \alpha \cdot r(S^*; \boldsymbol{\mu}) - r(S; \boldsymbol{\mu}))$. For each arm i, we define*

$$\Delta^i_{\min} = \inf_{S \in \mathcal{S}: p_i^{D,S} > 0, \Delta_S > 0} \Delta_S, \qquad \Delta^i_{\max} = \sup_{S \in \mathcal{S}: p_i^{D,S} > 0, \Delta_S > 0} \Delta_S.$$

As a convention, if there is no action S such that $p_i^{D,S} > 0$ and $\Delta_S > 0$, we define $\Delta^i_{\min} = +\infty$, $\Delta^i_{\max} = 0$. We define $\Delta_{\min} = \min_{i \in [m]} \Delta^i_{\min}$, and $\Delta_{\max} = \max_{i \in [m]} \Delta^i_{\max}$.

Let $\tilde{S} = \{i \in [m] \mid p_i^{D,S} > 0\}$ be the set of arms that could be triggered by S. Let $K = \max_{S \in \mathcal{S}} |\tilde{S}|$. We use $\lceil x \rceil_0$ to denote $\max\{\lceil x \rceil, 0\}$ for any real number x.

Theorem 2.1. *For the CUCB algorithm on a CMAB-T problem instance that satisfies monotonicity (Condition 2.1) and TPM bounded smoothness (Condition 2.2) with bounded smoothness constant B, (1) if $\Delta_{\min} > 0$, we have distribution-dependent bound*

$$Reg(T; \alpha, \beta, \boldsymbol{\mu}) \le \sum_{i \in [m]} \frac{576 B^2 K \ln T}{\Delta^i_{\min}} + \sum_{i \in [m]} \left(\left\lceil \log_2 \frac{2BK}{\Delta^i_{\min}} \right\rceil_0 + 2 \right) \cdot \frac{\pi^2}{6} \cdot \Delta_{\max} + 4Bm;$$

(2) we have distribution-independent bound

$$Reg(T; \alpha, \beta, \boldsymbol{\mu}) \le 12B\sqrt{mKT \ln T} + \left(\left\lceil \log_2 \frac{T}{18 \ln T} \right\rceil_0 + 2 \right) \cdot m \cdot \frac{\pi^2}{6} \cdot \Delta_{\max} + 2Bm.$$

2.4 Batch-size Independent Regret Bound

As defined in the previous section, K is the maximum number of arms that can be triggered, which also has been considered as the batch size [18]. In this section, for

Algorithm 2.1 CUCB with computation oracle.

Input: m, Oracle
For each arm i, $T_i \leftarrow 0$ {maintain the total number of times arm i is played}
For each arm i, $\hat{\mu}_i \leftarrow 1$ {maintain the empirical mean of X_i}
for $t = 1, 2, 3, \ldots$ **do**
  For each arm $i \in [m]$, $\rho_i \leftarrow \sqrt{\frac{3 \ln t}{2T_i}}$ {the confidence radius, $\rho_i = +\infty$ if $T_i = 0$}
  For each arm $i \in [m]$, $\bar{\mu}_i = \min\{\hat{\mu}_i + \rho_i, 1\}$ {the upper confidence bound}
  $S \leftarrow \mathsf{Oracle}(\bar{\mu}_1, \ldots, \bar{\mu}_m)$
  Play action S, which triggers a set $\tau \subseteq [m]$ of base arms with feedback $X_i^{(t)}$'s, $i \in \tau$
  For every $i \in \tau$, update T_i and $\hat{\mu}_i$: $T_i = T_i + 1$, $\hat{\mu}_i = \hat{\mu}_i + (X_i^{(t)} - \hat{\mu}_i)/T_i$
end for

the CMAB-T framework, we improve the regret dependency on the batch size K from $O(K)$ in [3] to $O(\log K)$ or $O(\log^2 K)$ in [14]. Notice that K is quite large in some applications, e.g., K can be hundreds of thousands for influence maximization in a large social network. Our main tool is a new condition called *triggering probability and variance modulated (TPVM) bounded smoothness condition*, replacing the TPM condition (Condition 2.2). We will define the TPVM condition, compare it with the TPM condition and the Gini-smoothness condition of [18], and show our algorithm and regret analysis that utilizes this condition.

2.4.1 TPVM Bounded Smoothness Condition

In this section, we introduce a new smoothness condition for many important applications as follows.

Condition 2.3 (Directional TPVM Bounded Smoothness)**.** *We say that a CMAB-T problem instance satisfies the directional TPVM (B_v, B_1, λ)-bounded smoothness condition ($B_v, B_1 \geq 0, \lambda \geq 1$), if for any action $S \in \mathcal{S}$, any distribution $D, D' \in \mathcal{D}$ with mean vector $\boldsymbol{\mu}, \boldsymbol{\mu}' \in (0,1)^m$, for any non-negative $\boldsymbol{\zeta}, \boldsymbol{\eta} \in [0,1]^m$ s.t. $\boldsymbol{\mu}' = \boldsymbol{\mu} + \boldsymbol{\zeta} + \boldsymbol{\eta}$, we have*

$$|r(S; \boldsymbol{\mu}') - r(S; \boldsymbol{\mu})| \leq B_v \sqrt{\sum_{i \in [m]} (p_i^{D,S})^{\lambda} \frac{\zeta_i^2}{(1-\mu_i)\mu_i}} + B_1 \sum_{i \in [m]} p_i^{D,S} \eta_i. \tag{2.2}$$

Remark 1 (Intuition for Condition 2.3). Looking at Eq.2.2, if we ignore

the $(1-\mu_i)\mu_i$ term in the denominator and set $\lambda = 2$, the RHS of 2.2 becomes $B_v\sqrt{\sum_{i\in[m]}(p_i^{D,S})^2\zeta_i^2} + B_1\sum_{i\in[m]} p_i^{D,S}\eta_i$, which holds with $B_v = B_1\sqrt{K}$ by applying the Cauchy-Schwarz inequality to Condition 2.2. However, the regret upper bound following this modified 2.2 would not directly lead to the improvement in the regret due to the $\sqrt{K}$ factor in B_v. To deal with this issue, an important observation here is that for many applications, the reason that B_v is large is the reward changes abruptly when parameters μ_i approach 0 or 1. This motivates us to plug in the $1/(1-\mu_i)\mu_i$ term in 2.2 to enlarge the square root term when μ_i is close to 0 or 1, so that B_v can be as small as possible. On the other hand, notice that when μ_i approaches 0 or 1, the variance $V_i \leq (1-\mu_i)\mu_i$ is also very small, [1] so the estimation of μ_i should be quite accurate. Therefore, the gap ζ_i between our estimation and true value produces a variance-related term that cancels the $(1-\mu_i)\mu_i$ in the denominator. Since ζ_i in 2.2 is modulated by both triggering probability $p_i^{D,S}$ and inverse upper bound of the variance $1/(1-\mu_i)\mu_i$, we call Condition 2.3 the directional triggering probability and variance modulated (TPVM) condition for short, where the term "directional" is explained in the next remark. The exponent $\lambda \geq 1$ on the triggering probability gives flexibility to trade-off between the strength of the condition and the quantity of the regret bound: With a larger λ, we can obtain a smaller regret bound, while with a smaller λ, the condition is easier to satisfy and allows us to include more applications.

Remark 2 (On directional TPVM vs. undirectional TPVM). In the above definition, "directional" means that we have $\boldsymbol{\zeta}, \boldsymbol{\eta} \geq \mathbf{0}$ such that $\boldsymbol{\mu}' \geq \boldsymbol{\mu}$ in every dimension. This is weaker than the version of the undirectional TPVM condition, where $\boldsymbol{\zeta}, \boldsymbol{\eta} \in [-1,1]^m$, and the η_i in the right hand side of Eq.(2.2) is replaced with $|\eta_i|$. The reason we use the weaker version is that some applications only satisfy the weaker version. To differentiate, we use TPVM$_<$ when we refer to the directional TPVM condition.

Remark 3 (Relation between Conditions 2.2 and 2.3). First, when setting $\boldsymbol{\zeta}$ to $\mathbf{0}$, the directional TPVM condition degenerates to the directional TPM condition. However, Condition 2.2 is the undirectional TPM condition, which is typically stronger than its directional counterpart. Thus, in general Condition 2.3 does not imply Condition 2.2. Nevertheless, with some additional assumptions

[1] For bounded random variable $X \in [0,1]$ with mean μ_i, variance $V_i = \mathbb{E}[X^2] - \mathbb{E}[X]^2 \leq \mathbb{E}[X] - (\mathbb{E}[X])^2 \leq (1-\mu_i)\mu_i$, where the equality is achieved when X is a Bernoulli random variable.

Algorithm 2.2 BCUCB-T: Bernstein Combinatorial Upper Confidence Bound Algorithm for CMAB-T

Input: Base arms $[m]$, computation oracle ORACLE.
Initialize: For each arm i, $T_{0,i} \leftarrow 0, \hat{\mu}_{0,i} = 0, \hat{V}_{0,i} = 0$.
for $t = 1, ..., T$ **do**
  For arm i, compute $\rho_{t,i}$ according to 2.3 and set UCB value $\bar{\mu}_{t,i} = \min\{\hat{\mu}_{t-1,i} + \rho_{t,i}, 1\}$.
  $S_t = \text{ORACLE}(\bar{\mu}_{t,1}, ..., \bar{\mu}_{t,m})$.
  Play S_t, which triggers arms $\tau_t \subseteq [m]$ with outcome $X_{t,i}$'s, for $i \in \tau_t$.
  For every $i \in \tau_t$, update $T_{t,i} = T_{t-1,i} + 1$, $\hat{\mu}_{t,i} = \hat{\mu}_{t-1,i} + (X_{t,i} - \hat{\mu}_{t-1,i})/T_{t,i}$, $\hat{V}_{t,i} = \frac{T_{t-1,i}}{T_{t,i}} \left(\hat{V}_{t-1,i} + \frac{1}{T_{t,i}} (\hat{\mu}_{t-1,i} - X_{t,i})^2 \right)$.
end for

Condition 2.3 does imply Condition 2.2 with the same coefficient B_1. Conversely, by applying the Cauchy-Schwartz inequality, one can verify that if a reward function is TPM B_1-bounded smooth, then it is (directional) TPVM $(B_1\sqrt{K}/2, B_1, \lambda)$-bounded smooth for any $\lambda \leq 2$. For some applications, we are able to reduce their B_v coefficient from $B_1\sqrt{K}/2$ to a coefficient independent of K, leading to significant savings in the regret bound.

Remark 4 (Comparing with [18]). [18] introduce a Gini-smoothness condition to reduce the batch-size dependency for CMAB problems, which largely inspires our TPVM$_<$ condition. Their condition is specified in a differential form of the reward function, with parameters γ_∞ and γ_g. We emphasize that their original condition cannot handle the probabilistic triggering setting in CMAB-T. One natural extension is to incorporate triggering probability modulation into their differential form of Gini-smoothness. However, we found that the resulting TPM Gini-smoothness condition is not strong enough to guarantee desirable regret bounds (See Appendix). This motivates us to provide a new condition directly on the difference form $|r(S; \boldsymbol{\mu}') - r(S; \boldsymbol{\mu})|$, similar to the TPM condition in [3]. Our TPVM$_<$ condition (Condition 2.3) can be viewed as extending Lemma 6 of [18] to incorporate triggering probabilities and bound the difference form $|r(S; \boldsymbol{\mu}') - r(S; \boldsymbol{\mu})|$. Intuitively, B_1 and B_v correspond to γ_∞ and γ_g, respectively, but since they are for different forms of definitions, their numerical values may not exactly match one another.

2.4.2 BCUCB-T Algorithm and Regret Analysis

Our proposed algorithm BCUCB-T is a generalization of the BC-UCB algorithm [18] which originally solves the non-triggering CMAB problem. Algorithm 2.2 maintains the empirical estimate $\hat{\mu}_{t,i}$ and $\hat{V}_{t,i}$ for the true mean and the true variance of the base arm outcomes. To select the action S_t, it feeds the upper confidence bound $\bar{\mu}_i$ into the offline oracle, where $\bar{\mu}_i$ optimistically estimates the μ_i by a confidence interval $\rho_{t,i}$. Compared with the CUCB algorithm which uses confidence interval $\rho_{t,i} = \sqrt{\frac{3\log t}{2T_{t-1,i}}}$ for the CMAB-T problem, the novel part is the usage of empirical variance $\hat{V}_{t-1,i}$ to construct the following "variance-aware" confidence interval:

$$\rho_{t,i} = \sqrt{\frac{6\hat{V}_{t-1,i}\log t}{T_{t-1,i}}} + \frac{9\log t}{T_{t-1,i}} \tag{2.3}$$

This confidence interval leverages on the empirical Bernstein inequality instead of the Chernoff-Hoeffding inequality. For the first term in 2.3, $\hat{V}_{t-1,i}$ is approximately equal to the true variance $V_i \leq (1-\mu_i)\mu_i$ and this indicates the estimation of μ_i is more accurate when μ_i is close to 0 or 1, which will cancel out the $(1-\mu_i)\mu_i$ coefficient of the B_v term in Condition 2.3 as we discussed before. The second term of Eq.(2.3) is to compensate the usage of the empirical variance $\hat{V}_{t-1,i}$, rather than the true variance V_i which is unknown to the learner. We next give its regret bound.

Theorem 2.2. *For a CMAB-T problem instance $([m], \mathcal{S}, \mathcal{D}, D_{trig}, R)$ that satisfies monotonicity (Condition 2.1), and TPVM$_<$ bounded smoothness (Condition 2.3) with coefficient (B_v, B_1, λ),*

(1) if $\lambda > 1$, BCUCB-T (Algorithm 2.2) with an (α, β)-approximation oracle achieves an (α, β)-approximate regret bounded by

$$O\left(\sum_{i\in[m]} \frac{B_v^2 \log K \log T}{\Delta_i^{\min}} + \sum_{i\in[m]} B_1 \log^2\left(\frac{B_1 K}{\Delta_i^{\min}}\right)\log T\right); \tag{2.4}$$

(2) if $\lambda = 1$, BCUCB-T (Algorithm 2.2) with an (α, β)-approximation oracle achieves an (α, β)-approximate regret bounded by

$$O\left(\sum_{i\in[m]} \log\left(\frac{B_v K}{\Delta_i^{\min}}\right)\frac{B_v^2 \log K \log T}{\Delta_i^{\min}} + \sum_{i\in[m]} B_1 \log^2\left(\frac{B_1 K}{\Delta_i^{\min}}\right)\log T\right). \tag{2.5}$$

Remark 5 (Discussion for Regret Bounds). Looking at the above regret bounds, for $\lambda > 1$ and $\lambda = 1$, the leading terms are $O(\sum_{i=1}^{m} \frac{B_v^2 \log K \log T}{\Delta_i^{\min}})$ and $O(\sum_{i=1}^{m} (\log \frac{B_v K}{\Delta_i^{\min}}) \frac{B_v^2 \log K \log T}{\Delta_i^{\min}})$. When $B_v \geq B_1$ (which typically holds) and gaps are small (i.e., $\Delta_{\min}^i \leq 1/\log^2 K$), the dependencies over K are $O(\log K)$ and $O(\log^2 K)$, respectively. For the setting of CMAB-T, [3] is the closest work to our work, where the reward function satisfies Condition 2.1 and Condition 2.2 with coefficient B_1. As mentioned in Remark 3 in 2.4.1, their reward function trivially satisfies our Condition 2.3 with coefficient $(B_1\sqrt{K}/2, B_1, 2)$ so our work reproduces a bound of $O(\sum_{i\in[m]} \frac{B_1^2 K \log K \log T}{\Delta_i^{\min}})$, matching [3] up to a factor of $O(\log K)$. For applications that satisfy TPVM (or TPVM$_<$) condition with non-trivial B_v, i.e., $B_v = o(B_1\sqrt{K})$, our work improves their regret bounds up to a factor of $O(K/\log K)$. As for the lower bound, according to the lower bound results in [19], our regret bound is tight up to a factor of $O(\log^2 K)$ on the (degenerate) non-triggering CMAB case.

Proof ideas. Our proof uses some events to filter the total regret and then bound these event-filtered regrets separately. As will be shown in the supplementary material, the event that contributes to the leading regret is $E_t = \{\Delta_{S_t} \leq e_t(S_t)\}$, where the error term $e_t(S_t) = O(B_v\sqrt{\sum_{i\in\tilde{S}_t}(\frac{\log t}{T_{t-1,i}})(p_i^{D,S_t})^\lambda} + B_1 \sum_{i\in\tilde{S}_t}(\frac{\log t}{T_{t-1,i}})(p_i^{D,S_t}))$. To handle the probabilistic triggering, our key ingredient is to use the triggering probability group technique proposed in [3] in the definition of the above events. For the $\lambda = 1$ case, one new issue arises since the triggering probability group divides sub-optimal actions S into *infinite* geometrically separated bins $(1/2, 1], (1/4, 1/2]..., (2^{-j}, 2^{-j+1}), ...,$ over $p_i^{D,S}$, and the regret should be proportional to the number of bins (which are infinitely large). To handle this, we show that it suffices to consider the first $j \leq j_i^{\max} = O(\log \frac{B_v K}{\Delta_i^{\min}})$ bins (which is why Eq.(2.5) has this additional factor in the leading term) and the regret of other bins (with very small $p_i^{D,S}$) can be safely neglected. To bound the leading regret filtered by E_t as mentioned earlier, we use the reverse amortization trick from [3, 20] and adaptively allocate each arm's regret contribution (according to thresholds on the number of times arm i is triggered). Note that these thresholds are carefully chosen for the error term $e_t(S_t)$, since trivially following the thresholds in [3] would either yield no meaningful bound or suffer from additional $O(\log T)$ or $O(\log K)$ factors in the regret. As a by-product, one can also use our analysis to replace that of [18] and [7] (where similar error term $e_t(S_t)$ appears) to improve their bound by a factor of $O(\log K)$. □

2.5 Summary

In this chapter, we review the general formulation of the non-competitive CMAB problem and two standard conditions of the reward function to achieve meaningful regret bounds. We discuss the CUCB algorithm from the previous work as well as its regret bound. We then introduce a new TPVM bounded smoothness condition for the reward function to enable batch-size independent regret bounds. The proposed BCUCB-T algorithm achieves improved regret dependency on the batch size compared to the CUCB algorithm.

2.6 Proof

In this section, we provide detailed proofs for Theorem 2.2. For the structure of this section, we first introduce some useful tools in Section 2.6.1 that will be helpful for our analysis. Next, we transform the total regret to the regret terms filtered by some events in Section 2.6.2. Then we provide regret bounds for all these regret terms. For these regret terms, we give the proof for the leading regret term: the proof giving Theorem 2.2 that uses the reverse amortization trick (see Eq. (2.38) and Eq. (2.49)) are in Section 2.6.3. It is notable that this trick can be used to improve [7, 18, 21] in a similar way, owing to the fact that their error terms have the similar form as ours shown in Eq. (2.24) (except without triggering probability modulation).

2.6.1 Useful Concentration Bounds and Definitions

We use the following tail bound for the construction of the confidence radius and our analysis.

Lemma 2.1 (Empirical Bernstein Inequality [22]). *Let $(X_i)_{i\in[n]}$ be n i.i.d random variables with bounded support $[0,1]$ and mean $\mathbb{E}[X_i]=\mu$. Let $\hat{X}_n \triangleq \frac{1}{n}\sum_{i\in[n]} X_i$ and $\hat{V}_n \triangleq \frac{1}{n}\sum_{i\in[n]}(X_i-\hat{X}_n)^2$ be the empirical mean and empirical variance of $(X_i)_{i\in[n]}$. Then for any $n\in\mathbb{N}$ and $y>0$, it holds that*

$$\Pr\left[|\hat{X}_n-\mu| \geq \sqrt{\frac{2\hat{V}_n y}{n}}+\frac{3y}{n}\right] \leq 3e^{-y} \tag{2.6}$$

We use the following Bernstein Inequality to bound the difference between the empirical variance and the true variance.

Lemma 2.2 (Bernstein Inequality [23]). *Let $(X_i)_{i\in[n]}$ be n independent random variables in $[0,1]$ with mean $\mathbb{E}[X_i]=\mu$ and variance $Var[X_i] \triangleq \mathbb{E}[X_i^2]-(\mathbb{E}[X_i])^2 = V$. Then with probability $1-\delta$:*

$$\frac{1}{n}\sum_{i\in[n]} X_i \leq \mu + \frac{2\log 1/\delta}{3n} + \sqrt{\frac{2V\log 1/\delta}{n}}. \tag{2.7}$$

Similar to [3], we define the event-filtered regret, the triggering group, the counter, the nice triggering event and the nice sampling event to help our analysis.

Definition 2.2 (Event-Filtered Regret)**.** *For any series of events $(\mathcal{E}_t)_{t\in[T]}$ indexed by round number t, we define the $Reg^A_{\alpha,\boldsymbol{\mu}}(T,(\mathcal{E}_t)_{t\in[T]})$ as the regret filtered by events $(\mathcal{E}_t)_{t\in[T]}$, or the regret is only counted in t if $\mathcal{E}$ happens in t. Formally,*

$$Reg^A_{\alpha,\boldsymbol{\mu}}(T,(\mathcal{E}_t)_{t\in[T]}) = \mathbb{E}\left[\sum_{t\in[T]} \mathbb{I}(\mathcal{E}_t)(\alpha\cdot r(S^*;\boldsymbol{\mu}) - r(S_t;\boldsymbol{\mu}))\right]. \tag{2.8}$$

For simplicity, we will omit $A,\alpha,\boldsymbol{\mu},T$ and rewrite $Reg^A_{\alpha,\boldsymbol{\mu}}(T,(\mathcal{E}_t)_{t\in[T]})$ as $Reg(T,\mathcal{E}_t)$ when contexts are clear.

Definition 2.3 (Triggering Probability (TP) group)**.** *For any arm i and index j, define the triggering probability (TP) group (of actions) as*

$$\mathcal{S}^D_{i,j} = \{S\in\mathcal{S} : 2^{-j} < p_i^{D,S} \le 2^{-j+1}\}. \tag{2.9}$$

Notice $\{\mathcal{S}^D_{i,j}\}$ forms a partition of $\{S\in\mathcal{S} : p_i^{D,S}\}$.

Definition 2.4 (Counter)**.** *For each TP group $S_{i,j}$, we define a counter $N_{i,j}$ which is initialized to 0. In each round t, if the action S_t is chosen, then we update $N_{i,j}$ to $N_{i,j}+1$ for (i,j) that $S_t\in S^D_{i,j}$. We also denote $N_{i,j}$ at the end of round t as $N_{t,i,j}$. Formally, we have the following recursive equation to define $N_{t,i,j}$ as follows:*

$$N_{t,i,j} = \begin{cases} 0, & \text{if } t=0 \\ N_{t-1,i,j}+1, & \text{if } t>0 \text{ and } S_t\in S^D_{i,j} \\ N_{t-1,i,j}, & \text{otherwise.} \end{cases} \tag{2.10}$$

Definition 2.5 (Nice triggering event $\mathcal{N}^t_t$)**.** *Given a series integers $\{j_i^{\max}\}_{i\in[m]}$, we say that the triggering is nice at the beginning of round t, if for every triggered group identified by (i,j), as long as $\frac{6\ln t}{\frac{1}{3}N_{t-1,i,j}2^{-j}} \le 1$, there is $T_{t-1,i} \ge \frac{1}{3}N_{t-1,i,j}\cdot 2^{-j}$. We denote this event as $\mathcal{N}^t_t$.*

Lemma 2.3 (Appendix B.1, Lemma 4 [3])**.** *For a series of integers $(j_i^{\max})_{i\in[m]}$, we have $\Pr[\neg\mathcal{N}^t_t] \le \sum_{i\in[m]} j_i^{\max} t^{-2}$ for every round $t\in[T]$.*

Proof. We refer the readers to Lemma 4 in Appendix B.1 from [3] for detailed proofs. □

Definition 2.6. *We say that the sampling is nice at the beginning of round t if: (1)*

for every base arm $i \in [m]$, $|\hat{\mu}_{t-1,i} - \mu_i| \leq \rho_{t,i}$, *where* $\rho_{t,i} = \sqrt{\frac{6\hat{V}_{t-1,i}\log t}{T_{t-1,i}}} + \frac{9\log t}{T_{t-1,i}}$*; (2) for every base arm* $i \in [m]$, $\hat{V}_{t-1,i} \leq 2\mu_i(1-\mu_i) + \frac{3.5\log t}{T_{t-1,i}}$. *We denote such event as* $\mathcal{N}_t^s$.

The following lemma bounds the probability that $\mathcal{N}_t^s$ does not happen.

Lemma 2.4. *For each round* t, $\Pr[\neg\mathcal{N}_t^s] \leq 4mt^{-2}$.

Proof. Let $\mathcal{N}_t^{s,1}, \mathcal{N}_t^{s,2}$ be the event (1) and event (2), where $\mathcal{N}_t^s = \mathcal{N}_t^{s,1} \bigcap \mathcal{N}_t^{s,2}$. We first bound the probability that $\mathcal{N}_t^{s,1}$ does not happen, we have

$$\Pr[\neg\mathcal{N}_t^{s,1}] = \Pr\left[\exists i \in [m] \text{ s.t. } |\hat{\mu}_{t-1,i} - \mu_i| > \sqrt{\frac{6\hat{V}_{t-1,i}\log t}{T_{t-1,i}}} + \frac{9\log t}{T_{t-1,i}}\right] \tag{2.11}$$

$$\leq \sum_{i\in[m]}\sum_{\tau\in[t]} \Pr\left[|\hat{\mu}_{t-1,i} - \mu_i| > \sqrt{\frac{6\hat{V}_{t-1,i}\log t}{\tau}} + \frac{9\log t}{\tau}, T_{t-1,i} = \tau\right] \tag{2.12}$$

$$\leq 3mt^{-2}, \tag{2.13}$$

where Eq. (2.12) is due to the union bound over i, τ, Eq. (2.13) is due to Lemma 2.1 by setting $y = 3\log t$ and when $T_{t-1,i} = \tau, \hat{\mu}_{t-1,i}$ and $\hat{V}_{t-1,i}$ are the empirical mean and empirical variance of τ i.i.d random variables with mean μ_i.

We then bound the probability that second event $\mathcal{N}_t^{s,2}$ does not happen using the similar proof of [18, Eq. (7)]. Fix $T_{t-1,i} = \tau$ and consider $(Y_i^1, ..., Y_i^\tau)$, where $Y_i^k = (X_i^k - \mu_i)^2 \in [0,1]$ and X_i^k is the random outcome of the k-th i.i.d trial. Since X_i^k are independent across k, Y_i^k are independent across k as well. In this case, one can verify that $\hat{V}_{t-1,i} = \frac{1}{\tau}\sum_{k=1}^{\tau}(X_i^k - \mu_i)^2 - (\frac{1}{\tau}\sum_{k=1}^{\tau} X_i^k - \mu_i)^2 \leq \frac{1}{\tau}\sum_{k=1}^{\tau}(X_i^k - \mu_i)^2 = \frac{1}{\tau}\sum_{k=1}^{\tau} Y_i^k$; $\mathbb{E}[Y_i^k] = \mathbb{E}[(X_i^k)^2] - \mu_i^2 \leq \mathbb{E}[X_i^k]\cdot 1 - \mu_i^2 = (1-\mu_i)\mu_i$; and $\mathrm{Var}[Y_i] = \mathbb{E}[(Y_i^k)^2] - (\mathbb{E}[Y_i^k])^2 \leq \mathbb{E}[(Y_i^k)^2] \leq \mathbb{E}[Y_i^k] \leq (1-\mu_i)\mu_i$. By Lemma 2.2 over τ i.i.d random variable $(Y_i^k)_{k\in\tau}$, it holds with probability at least $1 - t^{-3}$ that

$$\frac{1}{\tau}\sum_{k=1}^{\tau} Y_i^k \leq \mathbb{E}[Y_i^k] + \frac{2\log t}{n} + \sqrt{\frac{6\mathrm{Var}[Y_i^k]\log t}{\tau}} \tag{2.14}$$

This implies

$$\hat{V}_{t-1,i} \le \frac{1}{\tau}\sum_{k=1}^{\tau} Y_i^k \le \mathbb{E}[Y_i^k] + \frac{2\log t}{\tau} + \sqrt{\frac{6\mathrm{Var}[Y_i^k]\log t}{\tau}} \tag{2.15}$$

$$\le \mu_i(1-\mu_i) + \frac{2\log t}{\tau} + \sqrt{\frac{6(1-\mu_i)\mu_i \log t}{\tau}} \tag{2.16}$$

$$\le \mu_i(1-\mu_i) + \frac{2\log t}{\tau} + \mu_i(1-\mu_i) + \frac{3\log t}{2\tau} \tag{2.17}$$

$$= 2\mu_i(1-\mu_i) + \frac{3.5\log t}{\tau} \tag{2.18}$$

where Eq. (2.16) is using $2ab \le a^2 + b^2$ and $a = \sqrt{2\mu_i(1-\mu_i)}, b = \sqrt{\frac{3\log t}{n}}$.

Now by applying union bound over $i \in [m]$ and $\tau \in [t]$, we have $\Pr[\neg\mathcal{N}_t^{s,2}] \le mt^{-2}$. Lastly, applying union bound over $\mathcal{N}_t^{s,1}$ and $\mathcal{N}_t^{s,2}$, we have $\Pr[\neg\mathcal{N}_t^{s}] \le 4mt^{-2}$. □

After setting up all the above definitions, we can prove Lemma 2.5 about the confidence radius, which appears in the main content.

Lemma 2.5. *Fix every base arm i and every time t, with probability at least $1-4mt^{-3}$, it holds that*

$$\mu_i \le \bar{\mu}_{t,i} \le \min\{\mu_i + 2\rho_{t,i}, 1\} \le \min\left\{\mu_i + 4\sqrt{3}\sqrt{\frac{\mu_i(1-\mu_i)\log t}{T_{t-1,i}}} + \frac{28\log t}{T_{t-1,i}}, 1\right\}. \tag{2.19}$$

Proof. Recall that $\bar{\mu}_{t,i} = \min\{\hat{\mu}_{t-1,i} + \rho_{t,i}, 1\} = \min\{\hat{\mu}_{t-1,i} + \sqrt{\frac{6\hat{V}_{t-1,i}\log t}{T_{t-1,i}}} + \frac{9\log t}{T_{t-1,i}}, 1\}$. Under event $N_t^{s,1}$, we have $|\mu_i - \hat{\mu}_{t,i}| \le \rho_{t,i}$ by Lemma 2.4, hence the first and the second inequality in Lemma 2.5 holds.

For the last inequality, under event $N_t^{s,2}$, it holds that

$$\mu_i + 2\rho_{t,i} = \mu_i + 2\left(\sqrt{\frac{6\hat{V}_{t-1,i}\log t}{T_{t-1,i}}} + \frac{9\log t}{T_{t-1,i}}\right) \tag{2.20}$$

$$\leq \mu_i + 2\left(\sqrt{\frac{6\cdot(2\mu_i(1-\mu_i)+\frac{3.5\log t}{T_{t-1,i}})\log t}{T_{t-1,i}}} + \frac{9\log t}{T_{t-1,i}}\right) \tag{2.21}$$

$$\leq \mu_i + 4\sqrt{3}\sqrt{\frac{\mu_i(1-\mu_i)\log t}{T_{t-1,i}}} + 2\sqrt{21}\frac{\log t}{T_{t-1,i}} + \frac{18\log t}{T_{t-1,i}} \tag{2.22}$$

$$\leq \mu_{t-1,i} + 4\sqrt{3}\sqrt{\frac{\mu_i(1-\mu_i)\log t}{T_{t-1,i}}} + \frac{28\log t}{T_{t-1,i}}, \tag{2.23}$$

where Eq. (2.22) uses $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$.

Since $\mathcal{N}_t^s = \mathcal{N}_t^{s,1} \bigcap \mathcal{N}_t^{s,2}$ and by Lemma 2.4, Eq. (2.19) holds with probability at least $1 - 4mt^{-2}$. □

2.6.2 Decompose the Total Regret to Event-Filtered Regrets

In this section, we decompose the regret

$$\begin{aligned} Reg(T,\{\}) &= Reg(T,\mathcal{N}_t^s,\mathcal{N}_t^o) + Reg(T,\neg(\mathcal{N}_t^s \bigcap \mathcal{N}_t^o)) \\ &\leq Reg(T,\mathcal{N}_t^s,\mathcal{N}_t^o) + Reg(T,\neg\mathcal{N}_t^s) + Reg(T,\neg\mathcal{N}_t^o), \end{aligned}$$

where $\mathcal{N}_t^s$ is defined in Definition 2.6, $\mathcal{N}_t^o$ denotes the event where oracle successfully outputs an α-approximate solution (with probability at least β). We have the following lemma to do the decomposition.

Lemma 2.6. *[Leading Regret Term] Let $r(S;\boldsymbol{\mu})$ be TPVM smoothness with coefficients (B_v, B_1, λ), and define the error term*

$$e_t(S_t) = 4\sqrt{3}B_v\sqrt{\sum_{i\in\tilde{S}_t}(\frac{\log t}{T_{t-1,i}} \wedge \frac{1}{28})(p_i^{D,S_t})^\lambda} + 28B_1\sum_{i\in\tilde{S}_t}(\frac{\log t}{T_{t-1,i}} \wedge \frac{1}{28})(p_i^{D,S_t}) \tag{2.24}$$

and event $E_t = \mathbb{I}\{\Delta_{S_t} \leq e_t(S_t)\}$. The regret of Algorithm 2.2, when used with (α,β)

approximation oracle is bounded by

$$Reg(T) \leq Reg(T, E_t) + \frac{2\pi^2}{3} m\Delta_{\max}. \tag{2.25}$$

Proof. Under event $\mathcal{N}_t^s, \mathcal{N}_t^o$, by Lemma 2.5, it is easily to check that

$$\begin{aligned}\bar{\mu}_{t,i} &\leq \min\{\mu_{t-1,i} + 4\sqrt{3}\sqrt{\frac{\mu_i(1-\mu_i)\log t}{T_{t-1,i}}} + \frac{28\log t}{T_{t-1,i}}, 1\} \\ &\leq \mu_{t-1,i} + 4\sqrt{3}\sqrt{\mu_i(1-\mu_i)(\frac{\log t}{T_{t-1,i}} \wedge \frac{1}{28})} + 28(\frac{\log t}{T_{t-1,i}} \wedge \frac{1}{28}) \end{aligned} \tag{2.26}$$

Therefore, it holds that

$$\alpha r(S^*; \boldsymbol{\mu}) \leq \alpha r(S^*; \bar{\boldsymbol{\mu}}_t) \leq r(S_t; \bar{\boldsymbol{\mu}}_t) \tag{2.27}$$

$$\leq r(S_t; \boldsymbol{\mu}) + 4\sqrt{3}B_v\sqrt{\sum_{i\in\tilde{S}_t}(\frac{\log t}{T_{t-1,i}} \wedge \frac{1}{28})(p_i^{D,S_t})^\lambda} + 28B_1\sum_{i\in\tilde{S}_t}(\frac{\log t}{T_{t-1,i}} \wedge \frac{1}{28})(p_i^{D,S_t}), \tag{2.28}$$

where the first inequality in Eq. (2.27) is due to monotonicity condition (Condition 2.1) and second inequality in Eq. (2.27) is due to event $\mathcal{N}_t^o$, Eq. (2.28) is because of Eq. (2.26) and the TPVM condition (Condition 2.3) by plugging in $\zeta_i = 4\sqrt{3}\sqrt{\mu_i(1-\mu_i)(\frac{\log t}{T_{t-1,i}} \wedge \frac{1}{28})}$ and $\eta_i = 28(\frac{\log t}{T_{t-1,i}} \wedge \frac{1}{28})$.

So $Reg(T, \mathcal{N}_t^s, \mathcal{N}_t^o) \leq Reg(T, E_t)$. Now for $Reg(T, \neg\mathcal{N}_t^s)$, by Lemma 2.4 it holds that

$$Reg(T, \neg\mathcal{N}_t^s) \leq \sum_{t=1}^{T} \Pr[\neg\mathcal{N}_t^s] \leq \sum_{t=1}^{T} 4mt^{-2} \leq \frac{2\pi^2}{3} m\Delta_{\max}. \tag{2.29}$$

Similarly by definition, it holds that

$$Reg(T, \neg\mathcal{N}_t^o) \leq (1-\beta)T\Delta_{\max}. \tag{2.30}$$

Therefore $Reg(T, \{\}) \leq Reg(T, E_t) + \frac{2\pi^2}{3} m\Delta_{\max} + (1-\beta)T\Delta_{\max}$. And we have $Reg(T) = Reg(T, \{\}) - (1-\beta)T\Delta\max \leq Reg(T, E_t) + \frac{2\pi^2}{3}\Delta_{\max}$, which concludes Lemma 2.6.

□

Recall that event $E_t = \{\Delta_{S_t} \leq e_t(S_t)\}$, where

$$e_t(S_t) = 4\sqrt{3}B_v\sqrt{\sum_{i\in\tilde{S}_t}(\frac{\log t}{T_{t-1,i}} \wedge \frac{1}{28})(p_i^{D,S_t})^\lambda} + 28B_1\sum_{i\in\tilde{S}_t}(\frac{\log t}{T_{t-1,i}} \wedge \frac{1}{28})(p_i^{D,S_t}).$$

We will further decompose the event-filtered regret $Reg(T, E_t)$ into two event-filtered regret $Reg(T, E_{t,1})$ and $Reg(T, E_{t,2})$,

$$Reg(T, E_t) \leq Reg(T, E_{t,1}) + Reg(T, E_{t,2}), \tag{2.31}$$

where $E_{t,1} = \{\Delta_{S_t} \leq 2e_{t,1}(S_t)\}$, $E_{t,2} = \{\Delta_{S_t} \leq 2e_{t,2}(S_t)\}$, and

$$e_{t,1}(S_t) = 4\sqrt{3}B_v\sqrt{\sum_{i\in\tilde{S}_t}(\frac{\log t}{T_{t-1,i}} \wedge \frac{1}{28})(p_i^{D,S_t})^\lambda},$$

$$e_{t,2}(S_t) = 28B_1\sum_{i\in\tilde{S}_t}(\frac{\log t}{T_{t-1,i}} \wedge \frac{1}{28})(p_i^{D,S_t}).$$

The above inequality holds since the following facts: We can observe $e_{t,1}(S_t) + e_{t,2}(S_t) = e_t(S_t)$. From E_t, we know either $E_{t,1}$ holds or $E_{t,2}$ holds. So E_t implies that $1 \leq \mathbb{I}\{E_{t,1}\}+\mathbb{I}\{E_{t,2}\}$, and thus $\Delta_{S_t}\mathbb{I}\{E_t\} \leq \Delta_{S_t}\mathbb{I}\{E_{t,1}\}+\Delta_{S_t}\mathbb{I}\{E_{t,2}\}$, which concludes $Reg(T, E_t) \leq Reg(T, E_{t,1}) + Reg(T, E_{t,2})$. The next two sections will provide two different proofs for $Reg(T, E_{t,1}), Reg(T, E_{t,2})$ separately, where the second improves the first by a factor of $O(\log K)$.

2.6.3 Improved Analysis Using the Reverse Amortized Trick

In this section, we are going to bound the $Reg(T, E_{t,1})$ and $Reg(T, E_{t,2})$ separately under the event $\mathcal{N}_t^t$. The idea is to use a refined reverse amortization trick originated in [3] and to allocate the regret Δ_{S_t} to each base arm according to carefully designed thresholds. Note that it is highly non-trivial to derive the right thresholds and regret allocation strategy so that the K, T factors are as small as possible, which is our main contribution.

Upper bound for $Reg(T, E_{t,1})$. We first break $Reg(T, E_{t,1})$ into two parts and bound them separately: $Reg(T, \neg\mathcal{N}_t^t)$ and $Reg(T, E_{t,1} \bigcap \mathcal{N}_t^t)$.

For $Reg(T, E_{t,1} \bigcap \mathcal{N}_t^t)$, under the event $\mathcal{N}_t^t$, let $c_1 = 4\sqrt{3}$ and we set $j_i^{\max} = \frac{1}{\lambda}(\lceil \log_2 \frac{c_1^2 B_v^2 K}{(\Delta_i^{\min})^2} \rceil + 1)$. We first define a regret allocation function

$$\kappa_{i,j,T}(\ell) = \begin{cases} \frac{c_1^2 B_v^2 2^{(-j+1)(\lambda-1)}}{\Delta_i^{\min}}, & \text{if } \ell = 0 \text{ and } j \le j_i^{\max}, \\ 2\sqrt{\frac{24 c_1^2 B_v^2 2^{(-j+1)(\lambda-1)} \log T}{\ell}}, & \text{if } 1 \le \ell \le L_{i,j,T,1} \text{ and } j \le j_i^{\max}, \\ \frac{48 c_1^2 B_v^2 2^{(-j+1)(\lambda-1)} \log T}{\Delta_i^{\min}} \frac{1}{\ell}, & \text{if } L_{i,j,T,1} < \ell \le L_{i,j,T,2} \text{ and } j \le j_i^{\max}, \\ 0, & \text{if } \ell > L_{i,j,T,2} \text{ or } j > j_i^{\max}, \end{cases} \tag{2.32}$$

where $L_{i,j,T,1} = \frac{24 c_1^2 B_v^2 2^{(-j+1)(\lambda-1)} \log T}{(\Delta_i^{\min})^2}$, $L_{i,j,T,2} = \frac{48 c_1^2 B_v^2 2^{(-j+1)(\lambda-1)} K \log T}{(\Delta_i^{\min})^2}$.

Lemma 2.7. *For any time $t \in [T]$, if N_t^t and $E_{t,1}$ hold, we have*

$$\Delta_{S_t} \le \sum_{i \in \tilde{S}_t} \kappa_{i, j_i^{S_t}, T}(N_{t-1, i, j_i^{S_t}}), \tag{2.33}$$

where $j_i^{S_t}$ is the index of the triggering group $S_{i,j}$ such that $2^{-j_i^{S_t}} < p_i^{D,S_t} \le 2^{-j_i^{S_t}+1}$.

Proof. By event $E_{t,1}$, which is defined in Eq. (2.24), we apply the reverse amortization

(Eq. (2.38))

$$\Delta_{S_t} \le \sum_{i\in\tilde{S}_t} \frac{4c_1^2 B_v^2 (p_i^{D,S_t})^\lambda \min\{\frac{\log t}{T_{i,t-1}}, \frac{1}{28}\}}{\Delta_{S_t}} \tag{2.34}$$

$$\le -\Delta_{S_t} + 2\sum_{i\in\tilde{S}_t} \frac{4c_1^2 B_v^2 (p_i^{D,S_t})^\lambda \min\{\frac{\log t}{T_{i,t-1}}, \frac{1}{28}\}}{\Delta_{S_t}} \tag{2.35}$$

$$\le \sum_{i\in\tilde{S}_t} \left(\frac{8c_1^2 B_v^2 (p_i^{D,S_t})^\lambda \min\{\frac{\log t}{T_{i,t-1}}, \frac{1}{28}\}}{\Delta_{S_t}} - \frac{\Delta_{S_t}}{|\tilde{S}_t|} \right) \tag{2.36}$$

$$\le \sum_{i\in\tilde{S}_t} \left(\frac{8c_1^2 B_v^2 (p_i^{D,S_t})^\lambda \min\{\frac{\log t}{\frac{1}{3}N_{t-1,i,j_i^{S_t}} 2^{-j_i^{S_t}}}, \frac{1}{28}\}}{\Delta_{S_t}} - \frac{\Delta_{S_t}}{|\tilde{S}_t|} \right) \tag{2.37}$$

$$\le \sum_{i\in\tilde{S}_t} \underbrace{\left(\frac{8c_1^2 B_v^2 (2^{-j_i^{S_t}+1})^\lambda \min\{\frac{\log t}{\frac{1}{3}N_{t-1,i,j_i^{S_t}} 2^{-j_i^{S_t}}}, \frac{1}{28}\}}{\Delta_{S_t}} - \frac{\Delta_{S_t}}{K} \right)}_{(2.38,i)}, \tag{2.38}$$

where Eq. (2.34) is by the definition of $E_{t,1}$, which says

$$\Delta_{S_t}^2 \le \sum_{i\in\tilde{S}_t} 4c_1^2 B_v^2 (p_i^{D,S_t})^\lambda \min\{\frac{\log t}{T_{i,t-1}}, \frac{1}{28}\},$$

by dividing both sides by $\Delta_{S_t} > 0$. Eq. (2.35) is because we double the LHS and RHS of Eq. (2.34) at the same time and then put one into the RHS, Eq. (2.36) is by putting $-\Delta_{S_t}$ inside the summation, Eq. (2.38) is due to $p_i^{D,S_t} \le 2^{-j_i^{S_t}+1}$ given by the definition of $j_i^{S_t}$ and $|\tilde{S}_t| \le K$.

Note that the Eq. (2.35) is called the reverse amortization trick, since we allocate two times of the total regret and then minus the Δ_{S_t} term to amortize the regret when $\ell > L_{i,j,T,2}$ or $j > j_i^{\max}$ in Eq. (2.33), which saves the analysis for arms that are sufficiently triggered. Now we bound (2.38, i) under different cases.

When $j > j_i^{\max}$, we have $(2.38, i) \le \frac{8c_1^2 B_v^2 (2^{-j_i^{S_t}+1})^\lambda}{\Delta_{S_t}} \cdot \frac{1}{28} - \frac{\Delta_{S_t}}{K} \le \frac{8c_1^2 B_v^2}{\Delta_{S_t}} \frac{(\Delta_i^{\min})^2}{c_1^2 B_v^2 K} \cdot \frac{1}{28} - \frac{\Delta_{S_t}}{K} \le \frac{\Delta_i^{\min}}{K} \cdot \frac{8}{28} - \frac{\Delta_{S_t}}{K} \le 0 = \kappa_{i,j_i^{S_t},T}(N_{t-1,i,j_i^{S_t}})$.

When $N_{t-1,i,j_i^{S_t}} > L_{i,j_i^{S_t},T,2}$**,** we have $(2.38,i) \leq \frac{8c_1^2B_v^2(2^{-j_i^{S_t}+1})^\lambda \log t}{\frac{1}{3}N_{t-1,i,j_i^{S_t}} \cdot 2^{-j_i^{S_t}}\Delta_{S_t}} - \frac{\Delta_{S_t}}{K} \leq \frac{48c_1^2B_v^2 2^{(-j_i^{S_t}+1)(\lambda-1)}\log T}{\Delta_{S_t}}\frac{1}{N_{t-1,i,j_i^{S_t}}} - \frac{\Delta_{S_t}}{K} < \frac{(\Delta_i^{\min})^2}{K\Delta_{S_t}} - \frac{\Delta_{S_t}}{K} \leq 0 = \kappa_{i,j_i^{S_t},T}(N_{t-1,i,j_i^{S_t}})$.

When $L_{i,j_i^{S_t},T,1} < N_{t-1,i,j_i^{S_t}} \leq L_{i,j_i^{S_t},T,2}$ **and** $j \leq j_i^{\max}$**,** we have $(2.38,i) \leq \frac{8c_1^2B_v^2(2^{-j_i^{S_t}+1})^\lambda \log t}{\frac{1}{3}N_{t-1,i,j_i^{S_t}}2^{-j_i^{S_t}}\Delta_{S_t}} - \frac{\Delta_{S_t}}{K} \leq \frac{48c_1^2B_v^2 2^{(-j_i^{S_t}+1)(\lambda-1)}\log T}{\Delta_{S_t}}\frac{1}{N_{t-1,i,j_i^{S_t}}} - \frac{\Delta_{S_t}}{K} < \frac{48c_1^2B_v^2 2^{(-j_i^{S_t}+1)(\lambda-1)}\log T}{\Delta_{S_t}}\frac{1}{N_{t-1,i,j_i^{S_t}}} \leq \frac{48c_1^2B_v^2 2^{(-j_i^{S_t}+1)(\lambda-1)}\log T}{\Delta_i^{\min}}\frac{1}{N_{t-1,i,j_i^{S_t}}} = \kappa_{i,j_i^{S_t},T}(N_{t-1,i,j_i^{S_t}})$.

When $N_{t-1,i,j_i^{S_t}} \leq L_{i,j_i^{S_t},T,1}$ **and** $j \leq j_i^{\max}$**,** we further consider two different cases $N_{t-1,i,j_i^{S_t}} \leq \frac{24c_1^2B_v^2 2^{(-j_i^{S_t}+1)(\lambda-1)}\log T}{(\Delta_{S_t})^2}$ or $\frac{24c_1^2B_v^2 2^{(-j_i^{S_t}+1)(\lambda-1)}\log T}{(\Delta_{S_t})^2} < N_{t-1,i,j_i^{S_t}} \leq L_{i,j_i^{S_t},T,1} = \frac{24c_1^2B_v^2 2^{(-j_i^{S_t}+1)(\lambda-1)}\log T}{(\Delta_i^{\min})^2}$.

For the former case, if there exists $i \in \tilde{S}_t$ so that $N_{t-1,i,j_i^{S_t}} \leq \frac{24c_1^2B_v^2 2^{(-j_i^{S_t}+1)(\lambda-1)}\log T}{(\Delta_{S_t})^2}$, then we know $\sum_{q\in\tilde{S}_t}\kappa_{i,j_q^{S_t},T}(N_{t-1,q,j_q^{S_t}}) \geq \kappa_{i,j_i^{S_t},T}(N_{t-1,i,j_i^{S_t}}) = 2\sqrt{\frac{24c_1^2B_v^2 2^{(-j_i^{S_t}+1)(\lambda-1)}\log T}{N_{t-1,i,j_i^{S_t}}}} \geq 2\Delta_{S_t} > \Delta_{S_t}$, which makes Eq. (2.33) holds no matter what. This means we do not need to consider this case for good.

For the later case, when $\frac{24c_1^2B_v^2 2^{(-j_i^{S_t}+1)(\lambda-1)}\log T}{(\Delta_{S_t})^2} < N_{t-1,i,j_i^{S_t}}$, we know that

$$
\begin{aligned}
(2.38,i) &\leq \frac{48c_1^2B_v^2 2^{(-j_i^{S_t}+1)(\lambda-1)}\log T}{\Delta_{S_t}}\frac{1}{N_{t-1,i,j_i^{S_t}}} \\
&= 2\sqrt{\frac{24c_1^2B_v^2 2^{(-j_i^{S_t}+1)(\lambda-1)}\log T}{(\Delta_{S_t})^2}\frac{1}{N_{t-1,i,j_i^{S_t}}}}\sqrt{\frac{24c_1^2B_v^2 2^{(-j_i^{S_t}+1)(\lambda-1)}\log T}{N_{t-1,i,j_i^{S_t}}}} \\
&\leq 2\sqrt{\frac{24c_1^2B_v^2 2^{(-j_i^{S_t}+1)(\lambda-1)}\log T}{N_{t-1,i,j_i^{S_t}}}} = \kappa_{i,j_i^{S_t},T}(N_{t-1,i,j_i^{S_t}}).
\end{aligned}
$$

When $\ell = 0$ **and** $j \leq j_i^{\max}$**,** we have $(2.38,i) \leq \frac{8c_1^2B_v^2(2^{-j_i^{S_t}+1})^\lambda}{\Delta_{S_t}} \cdot \frac{1}{28} - \frac{\Delta_{S_t}}{K} \leq \frac{c_1^2B_v^2(2^{-j_i^{S_t}+1})^\lambda}{\Delta_{S_t}} \leq \frac{c_1^2B_v^2(2^{-j_i^{S_t}+1})^\lambda}{\Delta_i^{\min}} = \kappa_{i,j_i^{S_t},T}(N_{t-1,i,j_i^{S_t}})$.

Combining all above cases, we have $\Delta_{S_t} \leq \sum_{i\in\tilde{S}_t}\kappa_{i,j_i^{S_t},T}(N_{t-1,i,j_i^{S_t}})$. □

Since $N_{t,i,j_i^{S_t}}$ is increased if and only if $i \in \tilde{S}_t$ and consider all possible N_{t,i,j_i^S}

where $\kappa_{i,j_i^S,T}(S, N_{t-1,i,j^S}) > 0$, we have

$$
\begin{aligned}
&Reg(T, E_{t,1} \bigcap N_t^t) \\
&\leq \sum_{t\in[T]} \sum_{i\in\tilde{S}_t} \kappa_{i,j_i^{S_t},T}(N_{t-1,i,j^{S_t}}) && (2.39)\\
&\leq \sum_{i\in[m]} \sum_{j=1}^{j_i^{\max}} \frac{c_1^2 B_v^2 (2^{-j+1})^\lambda}{\Delta_i^{\min}} + \sum_{i\in[m]} \sum_{j=1}^{j_i^{\max}} \sum_{\ell=1}^{L_{i,j,T,1}} 2\sqrt{\frac{24 c_1^2 B_v^2 2^{(-j+1)(\lambda-1)} \log T}{\ell}} \\
&+ \sum_{i\in[m]} \sum_{j=1}^{j_i^{\max}} \sum_{L_{i,j,T,1}+1}^{L_{i,j,T,2}} \frac{48 c_1^2 B_v^2 2^{(-j+1)(\lambda-1)} \log T}{\Delta_i^{\min}} \frac{1}{\ell} && (2.40)\\
&\leq \sum_{i\in[m]} \sum_{j=1}^{j_i^{\max}} \frac{c_1^2 B_v^2 (2^{-j+1})^\lambda}{\Delta_i^{\min}} + \sum_{i\in[m]} \sum_{j=1}^{j_i^{\max}} \frac{96 c_1^2 B_v^2 2^{(-j+1)(\lambda-1)} \log T}{\Delta_i^{\min}} \\
&+ \sum_{i\in[m]} \sum_{j=1}^{j_i^{\max}} \frac{48 c_1^2 B_v^2 2^{(-j+1)(\lambda-1)} \log T}{\Delta_i^{\min}} (1 + \log K) && (2.41)\\
&= \sum_{i\in[m]} \sum_{j=1}^{j_i^{\max}} \frac{c_1^2 B_v^2 (2^{-j+1})^\lambda}{\Delta_i^{\min}} + \sum_{i\in[m]} \sum_{j=1}^{j_i^{\max}} \frac{48 c_1^2 B_v^2 2^{(-j+1)(\lambda-1)} \log T}{\Delta_i^{\min}} (3 + \log K) && (2.42)
\end{aligned}
$$

When $\lambda > 1$, we have

$$
\begin{aligned}
Reg(T, E_{t,1} \bigcap N_t^t) &\leq \sum_{i\in[m]} \sum_{j=1}^{\infty} \frac{c_1^2 B_v^2 2^{-j+1}}{\Delta_i^{\min}} \\
&\quad + \sum_{i\in[m]} \sum_{j=1}^{\infty} \frac{48 c_1^2 B_v^2 2^{(-j+1)(\lambda-1)} \log T}{1 - 2^{(\lambda-1)} \Delta_i^{\min}} (3 + \log K) \\
&= \sum_{i\in[m]} \frac{2 c_1^2 B_v^2}{\Delta_i^{\min}} + \sum_{i\in[m]} \frac{48 c_1^2 B_v^2 \log T}{\Delta_i^{\min}} (3 + \log K).
\end{aligned}
$$

When $\lambda = 1$, we have

$$
\begin{aligned}
Reg(T, E_{t,1} \bigcap N_t^t) &\leq \sum_{i\in[m]} \sum_{j=1}^{\infty} \frac{c_1^2 B_v^2 2^{-j+1}}{\Delta_i^{\min}} + \sum_{i\in[m]} j_i^{\max} \frac{48 c_1^2 B_v^2 \log T}{\Delta_i^{\min}} (3 + \log K) \\
&= \sum_{i\in[m]} \frac{2 c_1^2 B_v^2}{\Delta_i^{\min}} + \sum_{i\in[m]} \log \frac{c_1^2 B_v^2 K}{(\Delta_i^{\min})^2} \frac{48 c_1^2 B_v^2 \log T}{\Delta_i^{\min}} (3 + \log K).
\end{aligned}
$$

When $\lambda > 1$, we have $Reg(T, E_{t,1}) \leq \sum_{i\in[m]} \frac{2c_1^2 B_v^2}{\Delta_i^{\min}} + \sum_{i\in[m]} \frac{48c_1^2 B_v^2 \log T}{\Delta_i^{\min}}(3+\log K) + \frac{m\pi^2}{6}\log_2\left(\frac{c_1^2 B_v^2 K}{\lambda(\Delta_{\min})^2}\right)\Delta_{\max}$.

When $\lambda = 1$, we have $Reg(T, E_{t,1}) \leq \sum_{i\in[m]} \frac{2c_1^2 B_v^2}{\Delta_i^{\min}} + \sum_{i\in[m]} \log\frac{c_1^2 B_v^2 K}{(\Delta_i^{\min})^2} \frac{48c_1^2 B_v^2 \log T}{\Delta_i^{\min}}(3+\log K) + \frac{m\pi^2}{6}\log_2\left(\frac{c_1^2 B_v^2 K}{(\Delta_{\min})^2}\right)\Delta_{\max}$.

Upper bound for $Reg(T, E_{t,2})$. As usual, we first break $Reg(T, E_{t,2})$ into two parts and bound them separately: $Reg(T, E_{t,2}\bigcap \mathcal{N}_t^t)$ and $Reg(T, \neg\mathcal{N}_t^t)$.

For $Reg(T, E_{t,2}\bigcap \mathcal{N}_t^t)$, under the event $\mathcal{N}_t^t$, let $c_2 = 28$ be a constant and $K = \max_{S\in\mathcal{S}} |\tilde{S}|$. We set $j_i^{\max} = \lceil \log_2 \frac{4B_1 c_2 K}{\Delta_i^{\min}} \rceil + 1$. We first define a regret allocation function

$$\kappa_{i,j,T}(\ell) = \begin{cases} \Delta_i^{\max}, & \text{if } 0 \leq \ell \leq L_{i,j,T,1} \text{ and } j \leq j_i^{\max} \\ \frac{24c_2 B_1 \log T}{\ell}, & \text{if } L_{i,j,T,1} < \ell \leq L_{i,j,T,2} \text{ and } j \leq j_i^{\max} \\ 0, & \text{if } \ell > L_{i,j,T,2} + 1 \text{ or } j > j_i^{\max}, \end{cases} \tag{2.43}$$

where $L_{i,j,T,1} = \frac{24c_2 B_1 \log T}{\Delta_i^{\max}}$, $L_{i,j,T,2} = \frac{24c_2 B_1 K \log T}{\Delta_i^{\min}}$.

Lemma 2.8. *For any time $t \in [T]$, if N_t^t and $E_{t,2}$ hold, we have*

$$\Delta_{S_t} \leq \sum_{i\in\tilde{S}_t} \kappa_{i,j_i^{S_t},T}(N_{t-1,i,j_i^{S_t}}), \tag{2.44}$$

where $j_i^{S_t}$ is the index of the triggering group $S_{i,j}$ such that $2^{-j_i^{S_t}} < p_i^{D,S_t} \leq 2^{-j_i^{S_t}+1}$.

Proof. By event $E_{t,2}$, we have

$$\Delta_{S_t} \le \sum_{i\in\tilde{S}_t} 2c_2B_1p_i^{D,S_t}\min\{\frac{\log t}{T_{i,t-1}},\frac{1}{28}\} \tag{2.45}$$

$$\le -\Delta_{S_t} + 2\sum_{i\in\tilde{S}_t} 2c_2B_1p_i^{D,S_t}\min\{\frac{\log t}{T_{i,t-1}},\frac{1}{28}\} \tag{2.46}$$

$$\le \sum_{i\in\tilde{S}_t}\left(4c_2B_1p_i^{D,S_t}\min\{\frac{\log t}{T_{i,t-1}},\frac{1}{28}\} - \frac{\Delta_{S_t}}{|\tilde{S}_t|}\right) \tag{2.47}$$

$$\le \sum_{i\in\tilde{S}_t}\left(4c_2B_1p_i^{D,S_t}\min\{\frac{\log t}{\frac{1}{3}N_{t-1,i,j_i^{S_t}}2^{-j_i^{S_t}}},\frac{1}{28}\} - \frac{\Delta_{S_t}}{|\tilde{S}_t|}\right) \tag{2.48}$$

$$\le \sum_{i\in\tilde{S}_t}\underbrace{\left(4c_2B_12^{-j_i^{S_t}+1}\min\{\frac{\log t}{\frac{1}{3}N_{t-1,i,j_i^{S_t}}2^{-j_i^{S_t}}},\frac{1}{28}\} - \frac{\Delta_{S_t}}{K}\right)}_{(2.49,i)}, \tag{2.49}$$

where Eq. (2.45) is by the definition of $E_{t,1}$ and by dividing both sides by $\Delta_{S_t} > 0$, Eq. (2.46) is because we double the LHS and RHS of Eq. (2.45) at the same time and then put one into the RHS, Eq. (2.47) is by putting $-\Delta_{S_t}$ inside the summation, Eq. (2.49) is due to $p_i^{D,S_t} \le 2^{-j_i^{S_t}+1}$ given by the definition of $j_i^{S_t}$ and $|\tilde{S}| \le K$.

Similar to Eq. (2.49), Eq. (2.46) is called the reverse amortization. Now we bound (2.49, i) under different cases.

When $j > j_i^{\max}$, we have $(2.49,i) \le 4c_2B_12^{-j_i^{S_t}+1} - \frac{\Delta_{S_t}}{K} \le 4c_2B_1\frac{\Delta_i^{\min}}{c_2B_1K} - \frac{\Delta_{S_t}}{K} \le \frac{\Delta_i^{\min}}{K}\frac{4}{28} - \frac{\Delta_{S_t}}{K} \le 0 = \kappa_{i,j_i^{S_t},T}(N_{t-1,i,j_i^{S_t}})$.

When $N_{t-1,i,j_i^{S_t}} > L_{i,j_i^{S_t},T,2}$**,** we have $(2.49,i) \le 4c_2B_12^{-j_i^{S_t}+1}\frac{\log t}{\frac{1}{3}N_{t-1,i,j_i^{S_t}}2^{-j_i^{S_t}}} - \frac{\Delta_{S_t}}{K} \le \frac{24c_2B_1\log T}{N_{t-1,i,j_i^{S_t}}} - \frac{\Delta_{S_t}}{K} < \frac{\Delta_i^{\min}}{K} - \frac{\Delta_{S_t}}{K} \le 0 = \kappa_{i,j_i^{S_t},T}(N_{t-1,i,j_i^{S_t}})$.

When $N_{t-1,i,j_i^{S_t}} \le L_{i,j_i^{S_t},T,2}$ **and** $j \le j_i^{\max}$**,** We have $(2.49,i) \le 4c_2B_12^{-j_i^{S_t}+1}\frac{\log t}{\frac{1}{3}N_{t-1,i,j_i^{S_t}}2^{-j_i^{S_t}}} - \frac{\Delta_{S_t}}{K} = \frac{24c_2B_1\log T}{N_{t-1,i,j_i^{S_t}}} - \frac{\Delta_{S_t}}{K} < \frac{24c_2B_1\log T}{N_{t-1,i,j_i^{S_t}}} = \kappa_{i,j_i^{S_t},T}(N_{t-1,i,j_i^{S_t}})$.

When $N_{t-1,i,j_i^{S_t}} \le L_{i,j_i^{S_t},T,1}$ **and** $j \le j_i^{\max}$**,** If there exists $i \in \tilde{S}_t$ so that $N_{t-1,i,j_i^{S_t}} \le L_{i,j_i^{S_t},T,1} =$, then we know $\sum_{q\in\tilde{S}_t}\kappa_{i,j_q^{S_t},T}(N_{t-1,q,j_q^{S_t}}) \ge \kappa_{i,j_i^{S_t},T}(N_{t-1,i,j_i^{S_t}}) = \Delta_i^{\max} \ge \Delta_{S_t}$, which makes Eq. (2.44) holds no matter what. This means we do not need to consider this case for good.

Combining all above cases, we have $\Delta_{S_t} \leq \sum_{i \in \tilde{S}_t} \kappa_{i,j_i^{S_t},T}(N_{t-1,i,j_i^{S_t}})$. $\square$

Since $N_{t,i,j_i^{S_t}}$ is increased if and only if $i \in \tilde{S}_t$ and consider all possible i, j_i^S and N_{t,i,j_i^S} where $\kappa_{i,j_i^S,T}(N_{t-1,i,j^S}) > 0$, we have

$$
\begin{aligned}
&Reg(T, E_{t,2} \bigcap N_t^t) \\
&\leq \sum_{t \in [T]} \sum_{i \in \tilde{S}_t} \kappa_{i,j_i^{S_t},T}(N_{t-1,i,j^{S_t}}) \\
&\leq \sum_{i \in [m]} \sum_{j=0}^{j_i^{\max}} \sum_{\ell=1}^{L_{i,j,T,1}} \Delta_i^{\max} + \sum_{i \in [m]} \sum_{j=1}^{j_i^{\max}} \sum_{L_{i,j,T,1}+1}^{L_{i,j,T,2}} \frac{24 c_2 B_1 \log T}{\ell} \\
&\leq \sum_{i \in [m]} \sum_{j=1}^{j_i^{\max}} 24 c_2 B_1 \log T + \sum_{i \in [m]} \sum_{j=1}^{j_i^{\max}} 24 c_2 B_1 \log(\frac{K \Delta_i^{\max}}{\Delta_i^{\min}}) \log T \\
&= \sum_{i \in [m]} \sum_{j=1}^{j_i^{\max}} 24 c_2 B_1 \left(1 + \log(\frac{K \Delta_i^{\max}}{\Delta_i^{\min}})\right) \log T \\
&\leq \sum_{i \in [m]} 24 c_2 B_1 \left(\log_2 \frac{B_1 c_2 K}{\Delta_i^{\min}}\right) \left(1 + \log(\frac{K \Delta_i^{\max}}{\Delta_i^{\min}})\right) \log T
\end{aligned}
$$

Finally we can bound $Reg(T, E_{t,2})$:

$$
\begin{aligned}
Reg(T, E_{t,2}) \leq &\sum_{i \in [m]} 24 c_2 B_1 \left(\log_2 \frac{B_1 c_2 K}{\Delta_i^{\min}}\right) \left(1 + \log(\frac{K \Delta_i^{\max}}{\Delta_i^{\min}})\right) \log T \\
&+ \frac{m \pi^2}{6} \log_2 \frac{4 B_1 c_2 K}{\Delta_i^{\min}} \Delta_{\max}.
\end{aligned}
$$

Chapter 3

Competitive CMAB from the Follower's Perspective

3.1 Introduction

In this chapter, we study the competitive CMAB problem from the follower's perspective, where a player and a competitor (or a group of competitors) play with the same set of arms. Playing on the same arm incurs competition, which might lead to a potential loss of the reward. We assume the follower can choose his action after observing the action of the competitor and consider how the follower can maximize his own reward given the competitor's actions. Figure 3.1 shows an example of it in coupon allocation. There are two items with coupons 1 and 2, respectively. If a customer receives a coupon, he will buy the corresponding item with an unknown probability. In real-world applications, the customers may need to buy the items repeatedly (e.g., buying mobile plans monthly), so it is possible to learn their preferences with the coupons and use them for future coupon allocations. The competition happens if a customer receives both coupons, which will affect the customer's decision. If we consider item 1 as the follower, with the known allocation of coupon 2, the problem becomes allocating coupon 1 to customers to maximize the total number of customers buying item 1. In the following sections, we will introduce the general problem formulation and corresponding algorithms, then discuss a concrete application to influence maximization in social networks.

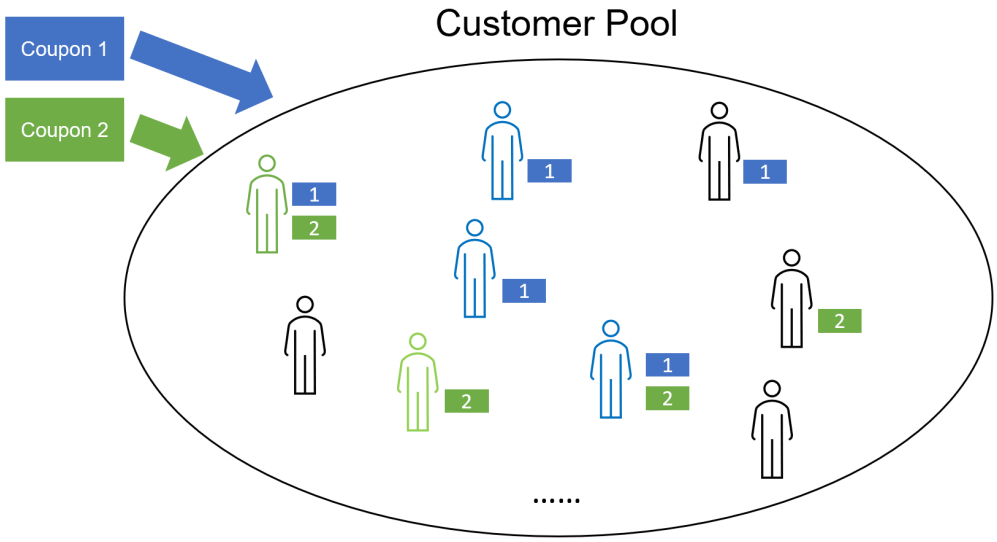


Figure 3.1: Competitive coupon allocation.

3.2 Problem Formulation

We consider a learning game with a player A and a competitor B.[1] We call player A the "follower" who will always choose actions after competitor B. They play with an environment consisting of m random variables $X_1, \cdots, X_m$ called *base arms* following a joint distribution D over $[0,1]^m$. Distribution D is chosen by the environment from a class of distributions $\mathcal{D}$ before the game starts. The player knows $\mathcal{D}$ but not the actual distribution D in advance.

The learning process runs in discrete rounds for $t = 1, 2, \ldots, T$. In round t, competitor B first takes an action $S_B^{(t)}$ from an action space $\mathcal{S}_B$, which is observed by follower A. Follower A then chooses an action $S_A^{(t)}$ from an action space $\mathcal{S}_A$, based on $S_B^{(t)}$ and the feedback history from previous rounds. Since the action of the competitor and the action of the follower will jointly affect the obtained reward of the follower, we define $S^{(t)} = (S_A^{(t)}, S_B^{(t)})$ as the joint action in round t and $\boldsymbol{\mathcal{S}}^{(t)} = \left\{ S \mid S = (S_A^{(t)}, S_B^{(t)}), S_A^{(t)} \in \mathcal{S}_A \right\}$ as the joint action space such that $S^{(t)} \in \boldsymbol{\mathcal{S}}^{(t)}$. Since the joint action space, $\boldsymbol{\mathcal{S}}^{(t)}$, can also be viewed as the context in round t, we call this setting as the general contextual combinatorial multi-armed bandit problem with probabilistically triggered arms (C^{2}MAB-T). The environment draws an independent sample $X^{(t)} = (X_1^{(t)}, \cdots, X_m^{(t)})$ from the joint distribution D. When action $S^{(t)}$ is

[1] It can be extended to multiple competitors by grouping their actions together.

played on the environment outcome $X^{(t)}$, a random subset of arms $\tau_t \in [m]$ are triggered, and the outcomes of $X_i^{(t)}$ for all $i \in \tau_t$ are observed as the feedback to the player. τ_t may have additional randomness beyond the randomness of $X^{(t)}$. Let $D_{\text{trig}}(S, X)$ denote a distribution of the triggered subset of $[m]$ for a given action S and an environment outcome X. We assume τ_t is drawn independently from $D_{\text{trig}}(S^{(t)}, X^{(t)})$. The player obtains a reward $R(S^{(t)}, X^{(t)}, \tau_t)$ fully determined by $S^{(t)}, X^{(t)}$ and τ_t. A learning algorithm aims at selecting actions $S^{(t)}$'s over time based on past feedback to accumulate as much reward as possible.

For each arm i, let $\mu_i = \mathbb{E}_{X \sim D}[X_i]$. Let $\boldsymbol{\mu} = (\mu_1, \cdots, \mu_m)$ denote the expectation vector of arms. We assume that the expected reward $\mathbb{E}[R(S, X, \tau)]$, where the expectation is taken over $X \sim D$ and $\tau \sim D_{\text{trig}}(S, X)$, is a function of action S and the expectation vector $\boldsymbol{\mu}$ of the arms. Thus, we denote $r_S(\boldsymbol{\mu}) := \mathbb{E}[R(S, X, \tau)]$. We assume the outcomes of arms do not depend on whether they are triggered, i.e., $\mathbb{E}_{X \sim D, \tau \sim D_{\text{trig}}(S,X)}[X_i \mid i \in \tau] = \mathbb{E}_{X \sim D}[X_i]$.

The performance of a learning algorithm $\mathcal{A}$ is measured by its expected regret, which is the difference in expected cumulative reward between always playing the best action and playing actions selected by algorithm $\mathcal{A}$. Let $\text{opt}^{(t)}(\boldsymbol{\mu}) = \sup_{S^{(t)} \in \boldsymbol{\mathcal{S}}^{(t)}} r_{S^{(t)}}(\boldsymbol{\mu})$ denote the expected reward of the optimal action in round t. We assume that there exists an offline oracle $\mathcal{O}$, which takes context $\boldsymbol{\mathcal{S}}^{(t)}$ and $\boldsymbol{\mu}$ as inputs and outputs an action $S^{\mathcal{O},(t)}$ such that $\Pr\{r_{S^{\mathcal{O},(t)}}(\boldsymbol{\mu}) \geq \alpha \cdot \text{opt}^{(t)}(\boldsymbol{\mu}))\} \geq \beta$, where α is the approximation ratio and β is the success probability. Instead of comparing with the exact optimal reward, we take the $\alpha\beta$ fraction of it and use the following (α, β)-approximation *frequentist regret* for T rounds:

$$Reg^{\mathcal{A}}_{\alpha,\beta}(T; \boldsymbol{\mu}) = \sum_{t=1}^{T} \alpha \cdot \beta \cdot \text{opt}^{(t)}(\boldsymbol{\mu}) - \sum_{t=1}^{T} r_{S^{\mathcal{A},(t)}}(\boldsymbol{\mu}), \tag{3.1}$$

where $S^{\mathcal{A},(t)}$ is the action chosen by algorithm $\mathcal{A}$ in round t.

Another way to measure the performance of the algorithm $\mathcal{A}$ is using *Bayesian regret*. Denote the prior distribution of $\boldsymbol{\mu}$ as $\mathcal{Q}$. When the prior $\mathcal{Q}$ is given, the corresponding Bayesian regret is defined as:

$$BayesReg^{\mathcal{A}}_{\alpha,\beta}(T) = \mathbb{E}_{\boldsymbol{\mu} \sim \mathcal{Q}} Reg^{\mathcal{A}}_{\alpha,\beta}(T; \boldsymbol{\mu}). \tag{3.2}$$

Note that the contextual combinatorial bandit problem is also studied in [24, 25].

They consider the context features of all bases arms, which can affect their expected outcomes in each round, and assume the action space of super arms is a subset of $[m]$. However, we do not bond the context with base arms and consider the feasible set of super arms, $\boldsymbol{\mathcal{S}}^{(t)}$, as the context, which is more flexible than a subset of $[m]$. Besides, we are the first to consider probabilistically triggered arms in the contextual combinatorial bandit problem.

In order to guarantee the theoretical regret bounds, we consider two conditions given in [3]. The first one is monotonicity, which is stated below.

Condition 3.1. *(Monotonicity). We say that a C²MAB-T problem instance satisfies monotonicity, if for any action S, for any two expectation vectors $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)$ and $\boldsymbol{\mu}' = (\mu'_1, \ldots, \mu'_m)$, we have $r_S(\boldsymbol{\mu}) \leq r_S(\boldsymbol{\mu}')$ if $\mu_i \leq \mu'_i$ for all $i \in [m]$.*

The second condition is Triggering Probability Modulated (TPM) Bounded Smoothness. We use $p_i^S(\boldsymbol{\mu})$ to denote the probability that the action S triggers arm i when the expectation vector is $\boldsymbol{\mu}$. The TPM condition in C²MAB-T is given below.

Condition 3.2. *(1-Norm TPM bounded smoothness). We say that a C²MAB-T problem instance satisfies 1-norm TPM bounded smoothness, if there exists $C \in \mathbb{R}^+$ (referred as the bounded smoothness coefficient) such that, for any two expectation vectors $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$, and any action S, $|r_S(\boldsymbol{\mu}) - r_S(\boldsymbol{\mu}')| \leq C \sum_{i \in [m]} p_i^S(\boldsymbol{\mu}) |\mu_i - \mu'_i|$.*

3.3 Algorithm and Regret Analysis

3.3.1 Algorithm with Monotonicity

For the general C²MAB-T problem that satisfies both monotonicity (Condition 3.1) and TPM bounded smoothness (Condition 3.2), we introduce a contextual version of the CUCB algorithm [3], which is described in Algorithm 3.1. Recall that $\boldsymbol{\mathcal{S}}^{(t)}$ is the action space in round t. We define the reward gap $\Delta_S^{(t)} = \max(0, \alpha \cdot \mathrm{opt}^{(t)}(\boldsymbol{\mu}) - r_S(\boldsymbol{\mu}))$ for all actions $S \in \boldsymbol{\mathcal{S}}^{(t)}$. For each arm i, we define $\Delta_{\min}^{i,T} = \min_{t \in [T]} \inf_{S \in \boldsymbol{\mathcal{S}}^{(t)} : p_i^S(\boldsymbol{\mu}) > 0, \Delta_S^{(t)} > 0} \Delta_S^{(t)}$ and $\Delta_{\max}^{i,T} = \max_{t \in [T]} \sup_{S \in \boldsymbol{\mathcal{S}}^{(t)} : p_i^S(\boldsymbol{\mu}) > 0, \Delta_S^{(t)} > 0} \Delta_S^{(t)}$. If there is no action S such that $p_i^S(\boldsymbol{\mu}) > 0$ and $\Delta_S^{(t)} > 0$, we define $\Delta_{\min}^{i,T} = +\infty$ and $\Delta_{\max}^{i,T} = 0$. We define $\Delta_{\min}^{(T)} = \min_{i \in [m]} \Delta_{\min}^{i,T}$ and $\Delta_{\max}^{(T)} = \max_{i \in [m]} \Delta_{\max}^{i,T}$. Let $\widetilde{S} = \{i \in [m] \mid p_i^S(\boldsymbol{\mu}) > 0\}$ be the set of arms that can be triggered by S. We define $K = \max_{S \in \boldsymbol{\mathcal{S}}^{(t)}} |\widetilde{S}|$ as the largest number of arms could be triggered by a feasible

Algorithm 3.1 Contextual CUCB with offline oracle $\mathcal{O}$, C^2-UCB

Input: m, Oracle $\mathcal{O}$.
For each arm $i \in [m]$, $T_i \leftarrow 0$. {maintain the total number of times arm i is played so far.}
For each arm $i \in [m]$, $\hat{\mu}_i \leftarrow 1$. {maintain the empirical mean of X_i.}
for $t = 1, 2, 3, \ldots$ **do**
    For each arm $i \in [m], \rho_i \leftarrow \sqrt{\frac{3 \ln t}{2T_i}}$. {the confidence radius, $\rho_i = +\infty$ if $T_i = 0$.}
    For each arm $i \in [m], \bar{\mu}_i = \min\{\hat{\mu}_i + \rho_i, 1\}$. {the upper confidence bound.}
    Obtain context $\boldsymbol{\mathcal{S}}^{(t)}$.
    $S^{(t)} \leftarrow \mathcal{O}(\boldsymbol{\mathcal{S}}^{(t)}, \bar{\mu}_1, \bar{\mu}_2, \ldots, \bar{\mu}_m)$.
    Play action $S^{(t)}$, which triggers a set $\tau \subseteq [m]$ of base arms with feedback $X_i^{(t)}$'s, $i \in \tau$.
    For every $i \in \tau$ update T_i and $\hat{\mu}_i$: $T_i = T_i + 1, \hat{\mu}_i = \hat{\mu}_i + (X_i^{(t)} - \hat{\mu}_i)/T_i$.
end for

action. We use $\lceil x \rceil_0$ to denote $\max\{\lceil x \rceil, 0\}$. Contextual CUCB (C^2-UCB) has the following regret bounds.

Theorem 3.1. *For the Contextual CUCB algorithm C^2-UCB (Algorithm 3.1) on an C^{2}MAB-T problem satisfying 1-norm TPM bounded smoothness (Condition 3.2) with bounded smoothness constant C, (1) if $\Delta_{\min}^{(T)} > 0$, we have a distribution-dependent bound*

$$Reg_{\alpha,\beta}(T; \boldsymbol{\mu}) \leq \sum_{i \in [m]} \frac{576 C^2 K \ln T}{\Delta_{\min}^{i,T}} + 4Cm + \sum_{i \in [m]} \left(\left\lceil \log_2 \frac{2CK}{\Delta_{\min}^{i,T}} \right\rceil_0 + 2 \right) \cdot \frac{\pi^2}{6} \cdot \Delta_{\max}^{(T)}, \tag{3.3}$$

and (2) we have a distribution-independent bound

$$Reg_{\alpha,\beta}(T; \boldsymbol{\mu}) \leq 12C\sqrt{mKT \ln T} + 2Cm + \left(\left\lceil \log_2 \frac{T}{18 \ln T} \right\rceil_0 + 2 \right) \cdot m \cdot \frac{\pi^2}{6} \cdot \Delta_{\max}^{(T)}.$$

Proof. We first show that Lemma 5 in [3] still holds for Contextual CUCB algorithm in the C^{2}MAB-T problem. Let $\mathcal{N}_t^{\mathrm{s}}$ be the event that at the beginning of round t, for every arm $i \in [m]$, $|\hat{\mu}_{i,t} - \mu_i| \leq \rho_{i,t}$. Let $\mathcal{H}_t$ be the event that at round t oracle $\mathcal{O}$ fails to output an α-approximate solution. In Lemma 5 from [3], it assumes that $\mathcal{N}_t^{\mathrm{s}}$ and $\neg\mathcal{H}_t$ hold, then we have

$$r_{S^{(t)}}(\bar{\boldsymbol{\mu}}_t) \geq \alpha \cdot \mathrm{opt}^{(t)}(\bar{\boldsymbol{\mu}}_t) \geq \alpha \cdot \mathrm{opt}^{(t)}(\boldsymbol{\mu}) = r_{S^{(t)}}(\boldsymbol{\mu}) + \Delta_{S^{(t)}}^{(t)}. \tag{3.4}$$

Algorithm 3.2 C^2-TS with offline oracle $\mathcal{O}$

Input: m, Prior $\mathcal{Q}$, Oracle $\mathcal{O}$.
Initialize Posterior $\mathcal{Q}_1 = \mathcal{Q}$
for $t = 1, 2, 3, \ldots$ **do**
  Draw a sample $\boldsymbol{\mu}^{(t)}$ from $\mathcal{Q}_t$.
  Obtain context $\boldsymbol{\mathcal{S}}^{(t)}$
  $S^{(t)} \leftarrow \mathcal{O}(\boldsymbol{\mathcal{S}}^{(t)}, \boldsymbol{\mu}^{(t)})$.
  Play action $S^{(t)}$, which triggers a set $\tau \subseteq [m]$ of base arms with feedback $X_i^{(t)}$'s, $i \in \tau$.
  Update posterior $\mathcal{Q}_{t+1}$ using $X_i^{(t)}$ for all $i \in \tau$.
end for

By the TPM condition, we have

$$\Delta_{S^{(t)}}^{(t)} \leq r_{S^{(t)}}(\bar{\boldsymbol{\mu}}_t) - r_{S^{(t)}}(\boldsymbol{\mu}) \leq C \sum_{i \in [m]} p_i^{S^{(t)}}(\boldsymbol{\mu}) |\bar{\mu}_{i,t} - \mu_i|, \tag{3.5}$$

which is in the same form of Eq.(10) in [3]. Hence, we can follow the remaining proof of its Lemma 5. With Lemma 5, we can follow the proof of Lemma 6 in [3] to bound the regret when $\Delta_{S^{(t)}}^{(t)} \geq M_{S^{(t)}}$, where $M_{S^{(t)}} = \max_{i \in \tilde{S}^{(t)}} M_i$ and M_i is a positive real number for each arm i. Finally, we take $M_i = \Delta_{\min}^{i,T}$. If $\Delta_{S^{(t)}}^{(t)} < M_{S^{(t)}}$, then $\Delta_{S^{(t)}}^{(t)} = 0$, since we have either $\tilde{S}^{(t)} = \emptyset$ or $\Delta_{S^{(t)}}^{(t)} < M_{S^{(t)}} \leq M_i$ for some $i \in \tilde{S}^{(t)}$. Thus, no regret is accumulated when $\Delta_{S^{(t)}}^{(t)} < M_{S^{(t)}}$. Following Eq.(17)-(22) in [3], we can derive the distribution-dependent and distribution-independent regret bounds shown in the theorem. □

3.3.2 Algorithm without Monotonicity

For the general C^{2}MAB-T problem without monotonicity, we proposed two algorithms, C^2-TS, C^2-OFU, that can still achieve logarithmic Bayesian and frequentist regrets respectively. We also present C^2-ETC that has a tradeoff between feedback requirement and regret bound.

C^2-TS is described in Algorithm 3.2. Different from OCIM-TS, we input a general prior $\mathcal{Q}$ (which depends on $\mathcal{D}$ and might not be Beta distributions anymore) and update the posterior distribution $\mathcal{Q}_t$ accordingly. With the same definitions in 3.3.1 and $\delta_{\max}^{(T)} = \max_{\boldsymbol{\mu}} \Delta_{\max}^{(T)}$, it has the following Bayesian regret bound.

Algorithm 3.3 C^2-OFU with offline oracle $\widetilde{\mathcal{O}}$

Input: m, Oracle $\widetilde{\mathcal{O}}$.
For each arm $i \in [m]$, $T_i \leftarrow 0$. {maintain the total number of times arm i is played so far.}
For each arm $i \in [m]$, $\hat{\mu}_i \leftarrow 1$. {maintain the empirical mean of X_i.}
for $t = 1, 2, 3, \ldots$ **do**
    For each arm $i \in [m], \rho_i \leftarrow \sqrt{\frac{3 \ln t}{2T_i}}$. {the confidence radius, $\rho_i = +\infty$ if $T_i = 0$.}
    For each arm $i \in [m], c_i \leftarrow [(\hat{\mu}_i - \rho_i)^{0+}, (\hat{\mu}_i + \rho_i)^{1-}]$. {the estimated range of μ_i.}
    Obtain context $\boldsymbol{\mathcal{S}}^{(t)}$.
    $S^{(t)} \leftarrow \widetilde{\mathcal{O}}(\boldsymbol{\mathcal{S}}^{(t)}, c_1, c_2, \ldots, c_m)$.
    Play action $S^{(t)}$, which triggers a set $\tau \subseteq [m]$ of base arms with feedback $X_i^{(t)}$'s, $i \in \tau$.
    For every $i \in \tau$ update T_i and $\hat{\mu}_i$: $T_i = T_i + 1, \hat{\mu}_i = \hat{\mu}_i + (X_i^{(t)} - \hat{\mu}_i)/T_i$.
end for

Theorem 3.2. *For the C^2-TS (Algorithm 3.2) on an C^{2}MAB-T problem satisfying 1-norm TPM bounded smoothness (Condition 3.2) with bounded smoothness constant C, we have the Bayesian regret bound*

$$BayesReg_{\alpha,\beta}(T) \leq 12C\sqrt{mKT \ln T} + 2Cm + \left(\left\lceil \log_2 \tfrac{T}{18 \ln T} \right\rceil_0 + 4\right) \cdot m \cdot \tfrac{\pi^2}{6} \cdot \delta_{\max}^{(T)}, \tag{3.6}$$

.

C^2-OFU is described in Algorithm 3.3. Similar to OCIM-OFU, it requires an offline oracle $\tilde{\mathcal{O}}$ that takes the context $\boldsymbol{\mathcal{S}}^{(t)}$ and c_i's (ranges of μ_i's) as inputs and outputs an approximate solution $S^{(t)}$. With such an oracle, C^2-OFU has the following frequentist regret bounds.

Theorem 3.3. *For the C^2-OFU (Algorithm 3.3) on an C^{2}MAB-T problem satisfying 1-norm TPM bounded smoothness (Condition 3.2) with bounded smoothness constant C, (1) if $\Delta_{\min}^{(T)} > 0$, we have a distribution-dependent bound*

$$Reg_{\alpha,\beta}(T; \boldsymbol{\mu}) \leq \sum_{i \in [m]} \tfrac{576C^2 K \ln T}{\Delta_{\min}^{i,T}} + 4Cm + \sum_{i \in [m]} \left(\left\lceil \log_2 \tfrac{2CK}{\Delta_{\min}^{i,T}} \right\rceil_0 + 2 \right) \cdot \tfrac{\pi^2}{6} \cdot \Delta_{\max}^{(T)}, \tag{3.7}$$

Algorithm 3.4 C^2-ETC with offline oracle $\mathcal{O}$

Input: m, k, N, T, Oracle $\mathcal{O}$.
For each arm i, $T_i \leftarrow 0$. {maintain the total number of times arm i is played.}
For each arm i, $\hat{\mu}_i \leftarrow 0$. {maintain the empirical mean of X_i.}
Exploration phase:
for $t = 1, 2, 3, \ldots, \lceil mN/k \rceil$ **do**
    Obtain context $\boldsymbol{\mathcal{S}}^{(t)}$.
    Play action $S^{(t)} \in \boldsymbol{\mathcal{S}}^{(t)}$, which contains k base arms that have not been chosen for N times.
    Observe the feedback $X_i^{(t)}$ for each base arm in $S^{(t)}$, $i \in \tau_{\text{direct}}$.
    For each arm $i \in \tau_{\text{direct}}$ update T_i and $\hat{\mu}_i$: $T_i = T_i + 1, \hat{\mu}_i = \hat{\mu}_i + (X_i^{(t)} - \hat{\mu}_i)/T_i$.
end for
Exploitation phase:
for $t = \lceil mN/k \rceil + 1, \ldots, T$ **do**
    Obtain context $\boldsymbol{\mathcal{S}}^{(t)}$.
    $S^{(t)} \leftarrow \mathcal{O}(\boldsymbol{\mathcal{S}}^{(t)}, \hat{\mu}_1, \hat{\mu}_2, \ldots, \hat{\mu}_m)$.
    Play action $S^{(t)}$.
end for

and (2) we have a distribution-independent bound

$$Reg_{\alpha,\beta}(T; \boldsymbol{\mu}) \leq 12C\sqrt{mKT \ln T} + 2Cm + \left(\left\lceil \log_2 \tfrac{T}{18 \ln T} \right\rceil_0 + 2\right) \cdot m \cdot \tfrac{\pi^2}{6} \cdot \Delta_{\max}^{(T)}.$$

Besides C^2-TS and C^2-OFU, we also provide a general explore-then-commit algorithm C^2-ETC, as described in Algorithm 3.4. In the general setting, τ_{direct} is defined as the set of base arms that is deterministically triggered by the action in question. C^2-ETC is simple and only requires feedback from directly triggered arms, but it has a worse regret bound and requires the following condition besides Condition 3.2.

Condition 3.3. *For some $k \geq 1$, given any context $\boldsymbol{\mathcal{S}}^{(t)}$ and any set $S' \subseteq [m]$ with $|S'| = k$, there exists $S \in \boldsymbol{\mathcal{S}}^{(t)}$ such that $p_i^S(\boldsymbol{\mu}) = 1$ for every $i \in |S'|$.*

With such a condition, C^2-ETC has the following frequentist regret bounds.

Theorem 3.4. *For the C^2-ETC (Algorithm 3.4) on an C^{2}MAB-T problem satisfying Condition 3.3 and 1-norm TPM bounded smoothness (Condition 3.2) with bounded smoothness constant C, (1) if $\Delta_{\min}^{(T)} > 0$, when $N = \max\left\{1, \frac{2C^2m^2}{(\Delta_{\min}^{(T)})^2} \ln(\frac{kT(\Delta_{\min}^{(T)})^2}{C^3 m})\right\}$,*

we have a distribution-dependent bound

$$Reg_{\alpha,\beta}(T;\boldsymbol{\mu}) \leq \frac{m}{k}\Delta_{\max}^{(T)} + \frac{2C^2m^3\Delta_{\max}^{(T)}}{k(\Delta_{\min}^{(T)})^2}\left(\max\left\{\ln\left(\frac{kT(\Delta_{\min}^{(T)})^2}{C^2m^2}\right),0\right\}+1\right) \tag{3.8}$$

and (2) when $N = (Ck)^{\frac{2}{3}}m^{-\frac{2}{3}}T^{\frac{2}{3}}(\ln T)^{\frac{1}{3}}$*, we have a distribution-independent bound*

$$Reg_{\alpha,\beta}(T;\boldsymbol{\mu}) \leq O(C^{\frac{2}{3}}m^{\frac{4}{3}}k^{-\frac{1}{3}}T^{\frac{2}{3}}(\ln T)^{\frac{1}{3}}). \tag{3.9}$$

The proofs of Theorem 3.2, 3.3 and 3.4 generally follow the same steps in Section 3.6.2, 3.6.3 and 3.6.5.

3.4 Application in Influence Maximization

3.4.1 Introduction

Influence maximization, motivated by viral marketing applications, has been extensively studied since [26] formally defined it as a stochastic optimization problem: given a social network G and a budget k, how should a set of k seed nodes in G be chosen such that the expected number of final activated nodes under a given diffusion model is maximized? They proposed the well-known Independent Cascade (IC) and Linear Threshold (LT) diffusion models, and gave a greedy algorithm that outputs a $(1-1/e-\epsilon)$-approximate solution for any $\epsilon > 0$. However, they only considered a single item (e.g., product, idea) propagating in the network. In reality, different items could propagate concurrently in the same network, interfering with each other and leading to competition during propagation. Several competitive diffusion models [27, 28, 29, 30, 31] have been proposed for this setting. We use a Competitive Independent Cascade (CIC) model [32], which extends the classical IC model to multi-item influence diffusion. We consider the competitive influence maximization problem between two items from the "follower's perspective": given the seed nodes of the competitor's item, the follower's item chooses a set of nodes so as to maximize the expected number of nodes activated by the follower's item, referred to as the influence spread of the item.

We refer to the above problem as "offline" competitive influence maximization, since the influence probabilities on edges, i.e., the probabilities of an item's propaga-

Table 3.1: Summary of the proposed algorithms.

Algorithm	No Prior?	Offline computation	Feedback	Regret
OCIM-TS	×	Standard	Full propagation	Bayes. $O(\sqrt{T \ln T})$
OCIM-OFU	✓	Hard	Full propagation	Freq. $O(\sqrt{T \ln T})$
OCIM-ETC	✓	Standard	Direct out-edges	Freq. $O(T^{\frac{2}{3}}(\ln T)^{\frac{1}{3}})$

tion along edges, are known in advance. It can be solved by a greedy algorithm due to submodularity [32]. However, in many real-world applications, the influence probabilities on edges are unknown. We study the competitive influence maximization in this setting, and call it the Online Competitive Influence Maximization (OCIM) problem. In OCIM, the influence probabilities on edges need to be learned through repeated influence maximization trials: in each round, given the seed nodes of the competitor, we (i) choose k seed nodes; (ii) observe the resulting diffusion that follows the CIC model to update our knowledge of the edge probabilities; and (iii) obtain a reward, which is the total number of nodes activated by our item. Our goal is to choose the seed nodes in each round based on previous observations so as to maximize the cumulative reward.

Most previous studies on the online non-competitive influence maximization problem use a combinatorial multi-armed bandit (CMAB) framework [2, 4], an extension of the classical multi-armed bandit problem that captures the tradeoff between exploration and exploitation in sequential decision making. In CMAB, a player chooses a combinatorial action to play in each round, observes a set of arms triggered by this action and receives a reward. The player aims to maximize her cumulative reward over multiple rounds, navigating a tradeoff between exploring unknown actions/arms and exploiting the best known action. CMAB algorithms must also deal with an exponential number of possible combinatorial actions, which makes exploring all actions infeasible.

To the best of our knowledge, we are the first to study the online competitive influence maximization problem. We introduce a general contextual combinatorial multi-armed bandit framework with probabilistically triggered arms (C^2MAB-T) for OCIM. Within this framework, OCIM presents a new challenge: the key monotonicity property (influence spread increases when influence probabilities on edges increase) no longer holds due to the competitive nature of propagation, and thus upper confidence bound (UCB) based algorithms [2, 4] cannot be directly applied to OCIM.

Such non-monotonicity also complicates the analysis of the important Triggering Probability Modulated (TPM) condition for CMAB [3], and we provide a non-trivial new proof to show it still holds for OCIM. We are the first to identify the OCIM problem as a natural CMAB problem without monotonicity and tackle it from three directions, providing three solutions with different tradeoffs, as shown in Table 3.1: OCIM-TS uses standard offline oracles to achieve good Bayesian regret, but requires prior knowledge of edge probabilities; OCIM-OFU has a stronger frequentist regret bound without prior knowledge, but requires harder offline computation; and OCIM-ETC uses standard offline oracles and fewer observations, but leads to a worse frequentist regret bound. None is a perfect solution for OCIM, but we believe their tradeoffs shed light on the challenges involved in solving OCIM and even general CMAB problems without monotonicity. Our regret analysis of OCIM-TS delicately combines the key property of Thompson Sampling (TS) with the TPM condition to tackle non-monotonicity and allows any benchmark (exact, approximate, or even heuristic) oracle; our analysis of OCIM-OFU and OCIM-ETC extends the analysis for CMAB to a new contextual setting (C^2MAB-T) where the contexts are defined as the feasible sets of super arms and are not bonded with base arms. Experiments on two real-world datasets demonstrate the effectiveness of our proposed algorithms.

Related Work. [26] formally defined the influence maximization problem in their seminal work. Since then, the problem has been extensively studied [33]. [34] presented a breakthrough approximation algorithm that runs in near-linear time, which was improved by a series of algorithms [35, 36, 37]. A number of studies [27, 28, 29, 30, 31, 38] addressed competitive influence maximization problems where multiple competing sources propagate in the same network. [27] proposed the distance-based and wave propagation models, and considered the influence maximization problem from the follower's perspective. [28] considered the CIC model and gave an algorithm for computing the best response to an opponent's strategy.

When the influence probabilities of edges are unknown, the non-competitive online influence maximization problem has been extensively studied [2, 3, 4, 7, 39, 40]. [2] studied the problem under the IC model and proposed a general CMAB framework. We introduce a new contextual extension of CMAB, called C^2MAB-T, different from the contextual CMAB studied by [24] and [25]: they consider the context features of all base arms and assume the action space of super arms is a subset of all base arms,

while we consider the feasible set of super arms as the context, which is more flexible than a subset of all base arms. [3] introduced a triggering probability modulated (TPM) bounded smoothness condition to remove an undesired factor in the regret bound of [2]. [7] introduced a budgeted online influence maximization framework, where marketers optimize their seed sets under a budget rather than a cardinality constraint. Our OCIM-TS algorithm is similar to the Combinatorial Thompson Sampling (CTS) algorithm of [20]. However, CTS requires an exact oracle and has frequentist regret bound, while OCIM-TS allows any benchmark oracle and has Bayesian regret bound. [41] studied the Bayesian regret of CTS for CMAB, but they also require an exact oracle and a monotonicity assumption that does not hold for OCIM. Our Bayesian regret analysis is also different from that of [42]: they only study a simple special CMAB problem, while we provide the regret bound for general C^{2}MAB-T instances, including the OCIM problem.

3.4.2 OCIM Formulation

In this section we present the formulation of OCIM. We first introduce the traditional competitive influence maximization problem, and then discuss its online extension where edge probabilities are unknown.

Competitive Independent Cascade Model

We consider a Competitive Independent Cascade (CIC) model, which is an extension of the classical IC model to multi-item influence diffusion. A network is modeled as a directed graph $G = (V, E)$ with $n = |V|$ nodes and $m = |E|$ edges. Every edge $(u, v) \in E$ is associated with a probability $p(u, v)$. There are two items, A and B, trying to propagate in G from their own seed sets S_A and S_B. The influence propagation runs as follows: nodes in S_A (resp. S_B) are activated by A (resp. B) at step 0; at each step $s \geq 1$, a node u activated by A (resp. B) in step $s-1$ tries to activate each of its inactive out-neighbors v to be A (resp. B) with an independent probability $p(u, v)$ that is the same for A and B (i.e., we consider a homogeneous CIC model). The homogeneity assumption is reasonable since typically A and B are two items of the same category (thus competing), so they are likely to have similar propagation characteristics.

If two in-neighbors of v activated by A and B respectively both successfully

activate v at step s, then a tie-breaking rule is applied at v to determine the final adoption. In this section, we consider two types of tie-breaking rules: dominance [29] and proportional [43] tie-breaking rules. Dominance tie-breaking with $A > B$ (resp. $B > A$) means v will always adopt A (resp. B) in a competition. Proportional tie-breaking means that if there are n_A in-neighbors activated by A and n_B in-neighbors activated by B trying to activate v at the same step, the probability that v adopts A (resp. B) is $\frac{n_A}{n_A+n_B}$ (resp. $\frac{n_B}{n_A+n_B}$). The same tie-breaking rule also applies to the case when a node u is selected both as an A-seed and a B-seed. The process stops when no nodes activated at a step s have inactive out-neighbors.

We consider the follower's perspective in the optimization task: let A be the follower and B be the competitor. Then given S_B, our goal is to choose at most k seed nodes in G as S_A to maximize the influence spread of A, denoted as $\sigma_A(S_A, S_B)$, which is the expected number of nodes activated by A after the propagation ends. According to [29]'s result, the above optimization task under the homogeneous CIC model with the dominance tie-breaking rule has the monotone and submodular properties, and thus can be approximately solved by a greedy algorithm.

OCIM Model

In the online competitive influence maximization (OCIM) problem, the edge probabilities $p(u, v)$'s are unknown and need to be learned: in each round t, given $S_B^{(t)}$, we can choose up to k seed nodes as $S_A^{(t)}$, observe the whole propagation of A and B that follows the CIC model, and obtain the reward, which is the number of nodes finally activated by A in this round. The propagation feedback observed is then used to update the estimates on edge probabilities $p(u, v)$'s, so that we can achieve better influence maximization results in subsequent rounds. Our goal is to accumulate as much reward as possible through this repeated process over multiple rounds.

We introduce a new contextual combinatorial multi-armed bandit framework with probabilistically triggered arms (C^2MAB-T) for the OCIM problem, which is a contextual extension of CMAB-T from [3]. In OCIM, the set of edges E is the set of (base) arms $[m] = \{1, ..., m\}$, and their outcomes follow m independent Bernoulli distributions with expectation $\mu_e = p(u, v)$ for all $e = (u, v) \in E$. We denote the independent samples of arms in round t as $X^{(t)} = (X_1^{(t)}, \ldots, X_m^{(t)}) \in \{0, 1\}^m$, where $X_i^{(t)} = 1$ means the i-th edge is on (or live) and $X_i^{(t)} = 0$ means the i-th edge is

off (or blocked) in round t, and thus $X^{(t)}$ corresponds to the *live-edge graph* [26] in round t. We consider the seed set of the competitor, $S_B^{(t)}$, as the *context* in round t since it is determined by the competitor and can affect our choice of $S_A^{(t)}$. We define $\boldsymbol{\mathcal{S}}^{(t)} = \left\{ S \mid S = (S_A^{(t)}, S_B^{(t)}), |S_A^{(t)}| \leq k \right\}$ as the action space in round t and $S^{(t)} \in \boldsymbol{\mathcal{S}}^{(t)}$ as the real action. We define the triggered arm set τ_t as the set of edges reached by the propagation from either $S_A^{(t)}$ or $S_B^{(t)}$. Thus, τ_t is the set of edges (u, v) where u can be reached from $S^{(t)}$ by passing through only edges $e \in E$ with $X_e^{(t)} = 1$. The outcomes of $X_i^{(t)}$ for all $i \in \tau_t$ are observed as the feedback. We denote the obtained reward in round t as $R(S^{(t)}, X^{(t)})$, which is the number of nodes finally activated by A. The expected reward $r_{S^{(t)}}(\boldsymbol{\mu}) = \mathbb{E}[R(S^{(t)}, X^{(t)})]$ is a function of the action $S^{(t)}$ and the vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)$. Note that our framework can also handle dynamic tie-breaking rules over different rounds, by treating the tie-breaking rule as a part of the context. For ease of explanation, we assume a fixed tie-breaking rule in this section.

The performance of a learning algorithm $\mathcal{A}$ is measured by its expected regret, which is the difference in expected cumulative reward between always playing the best action and playing actions selected by algorithm $\mathcal{A}$. Let $\mathrm{opt}^{(t)}(\boldsymbol{\mu}) = \sup_{S_A^{(t)}} r_{S^{(t)}}(\boldsymbol{\mu})$ denote the expected reward of the optimal action in round t. Since the offline influence maximization under the CIC model is NP-hard [29], we assume that there exists an offline (α, β)-approximation oracle $\mathcal{O}$, which takes $S_B^{(t)}$ and $\boldsymbol{\mu}$ as inputs and outputs an action $S^{\mathcal{O},(t)}$ such that $\Pr\{r_{S^{\mathcal{O},(t)}}(\boldsymbol{\mu}) \geq \alpha \cdot \mathrm{opt}^{(t)}(\boldsymbol{\mu}))\} \geq \beta$, where α is the approximation ratio and β is the success probability. Instead of comparing with the exact optimal reward, we use the following (α, β)-approximation *frequentist regret* for T rounds:

$$Reg_{\alpha,\beta}^{\mathcal{A}}(T; \boldsymbol{\mu}) = \sum_{t=1}^{T} \alpha \cdot \beta \cdot \mathrm{opt}^{(t)}(\boldsymbol{\mu}) - \sum_{t=1}^{T} r_{S^{\mathcal{A},(t)}}(\boldsymbol{\mu}), \tag{3.10}$$

where $S^{\mathcal{A},(t)} := (S_A^{\mathcal{A},(t)}, S_B^{(t)})$ is the action chosen by algorithm $\mathcal{A}$ in round t. Here $S_B^{(t)}$ is the context and $S_A^{\mathcal{A},(t)}$ is the seed set of item A chosen by algorithm $\mathcal{A}$.

Another way to measure the performance of the algorithm $\mathcal{A}$ is using *Bayesian regret* [44]. Denote the prior distribution of $\boldsymbol{\mu}$ as $\mathcal{Q}$ (we will discuss how to derive $\mathcal{Q}$ for OCIM in Section 3.4.4). When the prior $\mathcal{Q}$ is given, the corresponding Bayesian

regret is defined as:

$$BayesReg^{\mathcal{A}}_{\alpha,\beta}(T) = \mathbb{E}_{\boldsymbol{\mu}\sim\mathcal{Q}} Reg^{\mathcal{A}}_{\alpha,\beta}(T;\boldsymbol{\mu}). \tag{3.11}$$

We will design algorithms to solve the OCIM problem and bound their achieved Bayesian and frequentist regrets in Section 3.4.4 and Section 3.4.5, respectively.

3.4.3 Properties of OCIM

In this section, we first show that the key monotonicity property for CMAB does not hold in OCIM. We then prove that the important Triggering Probability Modulated (TPM) condition still holds, which is essential for the analysis of all proposed algorithms.

Non-monotonicity

The monotonicity condition given by [3] could be stated as follows in the context of OCIM: for any action $S = (S_A, S_B)$, for any two expectation vectors $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)$ and $\boldsymbol{\mu}' = (\mu'_1, \ldots, \mu'_m)$, we have $r_S(\boldsymbol{\mu}) \le r_S(\boldsymbol{\mu}')$ if $\mu_i \le \mu'_i$ for all $i \in [m]$. Figure 3.2 shows a simple example of OCIM that does not satisfy the monotonicity condition. The left and right nodes are the seed nodes of A and B; the numbers below edges are influence probabilities. It is easy to calculate that $r_S(\boldsymbol{\mu}) = \mu_1(1-\mu_2) + 2$, for both dominance and proportional tie-breaking rules. Thus, if we increase μ_2, $r_S(\boldsymbol{\mu})$ will decrease, which is contrary to monotonicity. In general, for every edge (u, v), depending on the positions of the A- and B-seeds, increasing the influence probability of (u, v) may benefit the propagation of A or may benefit the propagation of B and thus impair the propagation of A. Thus, the influence spread of A has intricate connections with the influence probabilities on the edges.

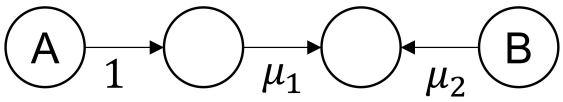


Figure 3.2: Example of non-monotonicity in OCIM.

The lack of monotonicity poses a significant challenge to the OCIM problem. We cannot directly use UCB-type algorithms [2], as they will not provide optimistic

solutions to bound the regret.

Triggering Probability Modulated (TPM) Bounded Smoothness

The lack of monotonicity further complicates the analysis of the Triggering Probability Modulated (TPM) condition [3], which is crucial in establishing regret bounds for CMAB algorithms. We use $p_i^S(\boldsymbol{\mu})$ to denote the probability that the action S triggers arm i when the expectation vector is $\boldsymbol{\mu}$. The TPM condition in OCIM is given below.

Condition 3.4. *(1-Norm TPM bounded smoothness). We say that an OCIM problem instance satisfies 1-norm TPM bounded smoothness, if there exists $C \in \mathbb{R}^+$ (referred to as the bounded smoothness coefficient) such that, for any two expectation vectors $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$, and any action $S = (S_A, S_B)$, we have $|r_S(\boldsymbol{\mu}) - r_S(\boldsymbol{\mu}')| \leq C \sum_{i\in[m]} p_i^S(\boldsymbol{\mu})|\mu_i - \mu_i'|$.*

Fortunately, with a more intricate analysis, we are able to show the following TPM condition.

Theorem 3.5. *Under both dominance and proportional tie-breaking rules, OCIM instances satisfy the 1-norm TPM bounded smoothness condition with coefficient $C = \tilde{C}$, where $\tilde{C}$ is the maximum number of nodes that any one node can reach in graph G.*

The proof of the above theorem is one of the key technical contributions of the paper. In the non-competitive setting, an edge coupling method could give a relatively simple proof for the TPM condition.[2] The idea of edge coupling is that for every edge $e \in E$, we sample a real number $X_e \in [0, 1]$ uniformly at random, and determine e to be live under $\boldsymbol{\mu}$ if $X_e \leq \mu_e$ and blocked if $X_e > \mu_e$, and similarly for $\boldsymbol{\mu}'$. This couples the live-edge graphs L and L' under $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$ respectively. In the non-competitive setting, due to the monotonicity property, we only need to consider the TPM condition when $\boldsymbol{\mu} \geq \boldsymbol{\mu}'$ (coordinate-wise), and this implies that L' is a subgraph of L, which significantly simplifies the analysis. However, in the competitive setting, monotonicity does not hold, and we have to show the TPM condition for every pair of $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$. Thus, L and L' no longer have the subgraph relationship. In this case, we have to show that for every coupling L and L', for

[2]The original proof in [3] occupies several pages, but [45] (in their Appendix E) provide a much shorter proof based on edge coupling.

Algorithm 3.5 OCIM-TS with offline oracle $\mathcal{O}$

Input: m, $\mathcal{O}$, Prior $\mathcal{Q} = \prod_{i\in[m]} Beta(a_i, b_i)$.
for $t = 1, 2, 3, \ldots$ **do**
  For each arm $i \in [m]$, draw a sample $\mu_i^{(t)}$ from $Beta(a_i, b_i)$; let $\boldsymbol{\mu}^{(t)} = (\mu_1^{(t)}, \cdots, \mu_m^{(t)})$.
  Obtain context $S_B^{(t)}$.
  $S^{(t)} \leftarrow \mathcal{O}(S_B^{(t)}, \boldsymbol{\mu}^{(t)})$.
  Play action $S^{(t)}$, which triggers a set $\tau \subseteq [m]$ of base arms with feedback $X_i^{(t)}$'s, $i \in \tau$.
  for all $i \in \tau$ **do**
    $a_i \leftarrow a_i + X_i^{(t)}; b_i \leftarrow b_i + 1 - X_i^{(t)}$.
  end for
end for

every $v \in V$ that is activated by A in L but not activated by A in L', it is because either (a) some edge $e = (u, w)$ is live in L but blocked in L' while u is A-activated (or equivalently e is A-triggered); or (b) some edge e is live in L' but blocked in L while e is B-triggered. The case (b) is due to the possibility of B blocking A's propagation, a unique scenario in OCIM. The above claim needs nontrivial inductive proofs for dominance and proportional tie-breaking rules, and then its correctness ensures the TPM condition.

3.4.4 Bayesian Regret Approach

In our OCIM model, since the samples of base arms follow Bernoulli distributions with mean vector $\boldsymbol{\mu}$, we can assume the prior distributions of $\boldsymbol{\mu}$, $\mathcal{Q}$, are Beta distributions, where $\mu_i \sim Beta(a_i, b_i)$ for all arm i. Given the prior distributions of all arms, we propose an Online Competitive Influence Maximization-Thompson Sampling (OCIM-TS) algorithm, which is described in Algorithm 3.5. We initialize the prior distribution of each arm i to $Beta(a_i, b_i)$. Then we take the context $S_B^{(t)}$ and the sampled $\boldsymbol{\mu}^{(t)}$ from prior distributions as inputs to the oracle $\mathcal{O}$, and get an output action $S^{(t)}$. After taking this action, we get feedback $X_i^{(t)}$'s from all triggered arms $i \in \tau$, then use them to update the prior distributions of all triggered base arms in τ. Let $\widetilde{S} = \{i \in [m] \mid p_i^S(\boldsymbol{\mu}) > 0\}$ be the set of arms that can be triggered by S. We define $K = \max_{S\in\boldsymbol{\mathcal{S}}^{(t)}} |\widetilde{S}|$ as the largest number of arms that could be triggered by a feasible action. We provide the Bayesian regret bound of OCIM-TS.

Theorem 3.6. *The OCIM-TS algorithm has the following Bayesian regret bound with $\tilde{C}$ as defined in Theorem 3.5:*

$$BayesReg_{\alpha,\beta}(T) \leq O(\tilde{C}\sqrt{mKT \ln T}). \tag{3.12}$$

This regret bound essentially matches the distribution-independent frequentist regret bound of OCIM-OFU in the next section. The proof of the above theorem is inspired by the posterior sampling regret decomposition of [44]. However, we combine the key property of posterior sampling with the TPM condition in Theorem 3.5 to tackle non-monotonicity. OCIM-TS can also be applied to general C^{2}MAB-T problems and allows any benchmark offline oracles (e.g., approximate or heuristic oracles).

3.4.5 Frequentist Regret Approach

Although OCIM-TS can solve the OCIM problem with a standard offline oracle (e.g., TCIM in [38]), it requires the prior distribution of the network parameter $\boldsymbol{\mu}$, which might not be available in practice. In this section, we first propose the OCIM-OFU algorithm. It achieves good frequentist regret without the prior knowledge, but requires a new oracle to solve a harder offline problem. We then design the OCIM-ETC algorithm, which requires less feedback and easier offline computation, but yields a worse frequentist regret bound.

OCIM-OFU Algorithm

As discussed in Section 3.4.3, due to the lack of monotonicity, we cannot directly use UCB-type algorithms. However, it is still possible to design bandit algorithms following the principle of Optimism in the Face of Uncertainty (OFU). We first introduce a new offline problem that jointly optimizes for both the seed set S^* and the optimal influence probability vector $\boldsymbol{\mu}^*$, where each dimension of $\boldsymbol{\mu}^*$, μ_i^*, is searched within a confidence interval c_i, for all $i \in E$.

$$\begin{aligned} \underset{S,\boldsymbol{\mu}}{\text{maximize}} \quad & r_S(\boldsymbol{\mu}) \\ \text{subject to} \quad & |S_A| \leq k, S = (S_A, S_B) \\ & \mu_i \in c_i,\ i = 1, \ldots, m. \end{aligned} \tag{3.13}$$

Algorithm 3.6 OCIM-OFU with offline oracle $\widetilde{\mathcal{O}}$

Input: m, Oracle $\widetilde{\mathcal{O}}$.
For each arm $i \in [m]$, $T_i \leftarrow 0$. {maintain the total number of times arm i is played so far.}
For each arm $i \in [m]$, $\hat{\mu}_i \leftarrow 1$. {maintain the empirical mean of X_i.}
for $t = 1, 2, 3, \ldots$ **do**
    For each arm $i \in [m], \rho_i \leftarrow \sqrt{\frac{3 \ln t}{2T_i}}$. {the confidence radius, $\rho_i = +\infty$ if $T_i = 0$.}
    For each arm $i \in [m], c_i \leftarrow [(\hat{\mu}_i - \rho_i)^{0+}, (\hat{\mu}_i + \rho_i)^{1-}]$. {the estimated range of μ_i.}
    Obtain context $S_B^{(t)}$.
    $S^{(t)} \leftarrow \widetilde{\mathcal{O}}(S_B^{(t)}, c_1, c_2, \ldots, c_m)$.
    Play action $S^{(t)}$, which triggers a set $\tau \subseteq [m]$ of base arms with feedback $X_i^{(t)}$'s, $i \in \tau$.
    For every $i \in \tau$ update T_i and $\hat{\mu}_i$: $T_i = T_i + 1, \hat{\mu}_i = \hat{\mu}_i + (X_i^{(t)} - \hat{\mu}_i)/T_i$.
end for

We then define a new offline (α, β)-approximation oracle $\widetilde{\mathcal{O}}$ to solve this problem. Oracle $\widetilde{\mathcal{O}}$ takes S_B and c_i's as inputs and outputs $\boldsymbol{\mu}^{\widetilde{\mathcal{O}}}$ and action $S^{\widetilde{\mathcal{O}}} = (S_A^{\widetilde{\mathcal{O}}}, S_B)$, such that $\Pr\{r_{S^{\widetilde{\mathcal{O}}}}(\boldsymbol{\mu}^{\widetilde{\mathcal{O}}}) \geq \alpha \cdot r_{S^*}(\boldsymbol{\mu}^*)\} \geq \beta$, where $(S^*, \boldsymbol{\mu}^*)$ is the optimal solution for Eq.(3.13).

With the offline oracle $\widetilde{\mathcal{O}}$, we propose an algorithm following the principle of Optimism in the Face of Uncertainty (OFU), named OCIM-OFU. The algorithm maintains the empirical mean $\hat{\mu}_i$ and confidence radius ρ_i for each edge probability. It uses the lower and upper confidence bounds to determine the range of μ_i: $c_i = [(\hat{\mu}_i - \rho_i)^{0+}, (\hat{\mu}_i + \rho_i)^{1-}]$, where we use $(x)^{0+}$ and $(x)^{1-}$ to denote $\max\{x, 0\}$ and $\min\{x, 1\}$ for any real number x. It feeds $S_B^{(t)}$ and all current c_i's into the offline oracle $\widetilde{\mathcal{O}}$ to obtain the action $S^{(t)} = (S_A^{(t)}, S_B^{(t)})$ to play at round t. The confidence radius ρ_i is large if arm i is not triggered often, which leads to a wider search space c_i to find the optimistic estimate of μ_i. We provide its frequentist regret bound.

Theorem 3.7. *The OCIM-OFU algorithm has the following distribution-independent bound with $\tilde{C}$ defined in Theorem 3.5 ,*

$$Reg_{\alpha,\beta}(T; \boldsymbol{\mu}) \leq O(\tilde{C}\sqrt{mKT \ln T})$$

The above regret bound has the typical form of $\sqrt{T \ln T}$, indicating that it is

tight on the important time horizon T. In fact, it has the same order as in [3]'s for the CMAB problem under monotonicity, despite the fact that the OCIM problem does not enjoy monotonicity, and matches the lower bound of CMAB with general reward functions in [19]. This result is due to our non-trivial TPM condition analysis (Theorem 3.5) that shows the same condition as in [3]'s setting with monotonicity.

Computational Efficiency. We now discuss the computational complexity of implementing the OCIM-OFU algorithm. We show the complexity of the new offline optimization problem in Eq. (3.13).

Theorem 3.8. *The offline problem in Eq.*(3.13) *is #P-hard.*

As mentioned before, the original offline problem, i.e., maximizing $r_S(\boldsymbol{\mu})$ over S when fixing $\boldsymbol{\mu}$, can be solved by several algorithms [38] based on submodularity of $r_S(\boldsymbol{\mu})$ over S. A straightforward attempt on the new offline problem in Eq.(3.13) is to show the submodularity of $g(S) = \max_{\boldsymbol{\mu}} r_S(\boldsymbol{\mu})$ over S, and then to use a greedy algorithm on g to select S. Unfortunately, we find that $g(S)$ is not submodular (see Section 3.6.4 for a counterexample). Implementing the oracle $\widetilde{\mathcal{O}}$ is then a challenge. However, it is possible to design efficient approximate oracles for bipartite graphs, which model the competitive probabilistic maximum coverage problem with applications in online advertising [2]. The main idea is that we can pre-determine that either the lower or the upper bound of c_i is optimal and should be chosen as μ_i^* depending on the tie-breaking rule, then use existing efficient influence maximization algorithms to get approximate solutions. The competitive propagation in the general graph is much more complicated, but we have a key observation that the optimal solution for the optimization problem in Eq.(3.13) must occur at the boundaries of the intervals c_i. Based on that, we discuss solutions for some specific graphs such as trees. See Section 3.6.4 for more details.

OCIM-ETC Algorithm

In this section, we propose an OCIM Explore-Then-Commit (OCIM-ETC) algorithm. It has two advantages: first, it does not need the new offline oracle discussed in Sec. 3.4.5; and second, it requires fewer observations than our other algorithms: instead of the observations of all triggered edges, i.e., τ, it only needs the observations of all direct out-edges of seed nodes.

Like other ETC algorithms [46], OCIM-ETC divides the T rounds into two phases:

Algorithm 3.7 OCIM-ETC with offline oracle $\mathcal{O}$

Input: m, N, T, Oracle $\mathcal{O}$.
For each arm i, $T_i \leftarrow 0$. {maintain the total number of times arm i is played so far.}
For each arm i, $\hat{\mu}_i \leftarrow 0$. {maintain the empirical mean of X_i.}
Exploration phase:
for $t = 1, 2, 3, \ldots, \lceil nN/k \rceil$ **do**
    Take k nodes that have not been chosen for N times as S_A.
    Observe the feedback $X_i^{(t)}$ for each direct out-edge of S_A, $i \in \tau_{\text{direct}}$.
    For each arm $i \in \tau_{\text{direct}}$ update T_i and $\hat{\mu}_i$: $T_i = T_i + 1, \hat{\mu}_i = \hat{\mu}_i + (X_i^{(t)} - \hat{\mu}_i)/T_i$.
end for
Exploitation phase:
for $t = \lceil nN/k \rceil + 1, \ldots, T$ **do**
    Obtain context $S_B^{(t)}$.
    $S^{(t)} \leftarrow \mathcal{O}(S_B^{(t)}, \hat{\mu}_1, \hat{\mu}_2, \ldots, \hat{\mu}_m)$.
    Play action $S^{(t)}$.
end for

an exploration phase and an exploitation phase. In the exploration phase, it chooses each node as the seed node of A for N times. The exploration phase thus takes $\lceil nN/k \rceil$ rounds. In the exploitation phase, it takes $S_B^{(t)}$ and the empirical means $\hat{\mu}_i$ as inputs to the oracle $\mathcal{O}$ mentioned in Sec. 3.4.2, then plays the output action $S^{\mathcal{O},(t)}$. We give its frequentist regret bound.

Theorem 3.9. *The OCIM-ETC algorithm has the following distribution-independent regret bound with $\tilde{C}$ defined in Theorem 3.5, when $N = (\tilde{C}mk)^{\frac{2}{3}} n^{-\frac{4}{3}} T^{\frac{2}{3}} (\ln T)^{\frac{1}{3}}$,*

$$Reg_{\alpha,\beta}(T; \boldsymbol{\mu}) \leq O((\tilde{C}mn)^{\frac{2}{3}} k^{-\frac{1}{3}} T^{\frac{2}{3}} (\ln T)^{\frac{1}{3}}). \tag{3.14}$$

Although this regret bound is worse than that of the OCIM-OFU algorithm in Theorem 3.7, OCIM-ETC requires easier offline computation and less feedback since it only needs to observe the results of direct out-edges of seed nodes, which shows the tradeoff between regret bound and feedback/computation in OCIM.

3.4.6 Extension to Probabilistic Competitor's Seed Distribution

In [38], the authors extend the offline CIM problem to a probabilistic setting where the competitor's seed distribution is known (i.e., the probability of each node being selected as a seed by the competitor). In this section, we extend our algorithms to handle two new settings where the competitor has a probabilistic seed distribution. Note that we need to slightly modify the TPM condition for these settings. We denote the expected reward of follower A as $r(S_A, D_B, \boldsymbol{\mu})$, where S_A is the seed set of A, D_B is the seed distribution of B. We use $p_i(S_A, D_B, \boldsymbol{\mu})$ to denote the probability that either S_A or S_B will trigger arm i when the seed set of A is S_A, the seed set of B, S_B, is sampled from D_B, and the expectation vector is $\boldsymbol{\mu}$. The modified TPM condition is given below.

Condition 3.5. *(Modified TPM bounded smoothness). We say that an OCIM problem instance satisfies modified TPM bounded smoothness, if there exists $C \in \mathbb{R}^+$ such that, for any two expectation vectors $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$, and any seed set S_A and seed distribution D_B, we have $|r(S_A, D_B, \boldsymbol{\mu}) - r(S_A, D_B, \boldsymbol{\mu}')| \leq C \sum_{i \in [m]} p_i(S_A, D_B, \boldsymbol{\mu}) |\mu_i - \mu_i'|$.*

With a similar analysis of Theorem 3.5, we can show the following TPM condition when the competitor has probabilistic seed distribution.

Theorem 3.10. *Under both dominance and proportional tie-breaking rules, OCIM instances satisfy the modified TPM bounded smoothness condition with coefficient $C = 2\tilde{C}$, where $\tilde{C}$ is the maximum number of nodes that any one node can reach in graph G.*

Known dynamic seed distribution. In round t, the competitor's seed set $S_B^{(t)}$ follows a distribution $D_B^{(t)}$, i.e., $S_B^{(t)} \sim D_B^{(t)}$. However, the follower only knows $D_B^{(t)}$ but not $S_B^{(t)}$ before choosing $S_A^{(t)}$. Since our proposed framework has a nice separation between online learning and offline computation, in this setting, only the offline computation part will be affected. Specifically, we can replace the oracle $\mathcal{O}(S_B^{(t)}, \boldsymbol{\mu}^{(t)})$ in OCIM-TS and OCIM-ETC with a new oracle $\mathcal{O}_{\text{new}}(D_B^{(t)}, \boldsymbol{\mu}^{(t)})$. For OCIM-OFU, similar to oracle $\widetilde{\mathcal{O}}$, we need a new oracle $\widetilde{\mathcal{O}}_{\text{new}}$ that takes $D_B^{(t)}$ and the confidence intervals $\{c_i\}$ as inputs and outputs $S_A^{(t)}$. We can use the TCIM algorithm of [38] to design $\mathcal{O}_{\text{new}}$ and $\widetilde{\mathcal{O}}_{\text{new}}$. Our proposed algorithms will have the same regret bounds as in Theorems 3.6 and 3.7.

Unknown fixed seed distribution. In this setting, the seed distribution of the

competitor, D_B, is unknown to the follower but fixed for all rounds. To solve this problem, we introduce a virtual B seed node u_B, which connects to each existing node u with an unknown edge probability $p(u_B, u)$ equal to the probability of u being selected as a B seed. This reduces the case of probabilistic seed selection to the standard CIC model with a known seed node u_B. The unknown edge probabilities $p(u_B, u)$'s can be learned together with the edge probabilities in the original graph. Therefore, we do not need to know the competitor's seed selection in advance and can learn it over time through the online learning process. Our algorithms will have the same regret guarantees as in Theorems 3.6 and 3.7.

3.4.7 Experiments

Datasets and settings. To validate our theoretical findings, we conduct experiments on two real-world datasets widely used in the influence maximization literature, with detailed statistics summarized in Table 3.2. First, we use the Yahoo! Search Marketing Advertiser Bidding Data[3] (denoted as Yahoo-Ad), which contains a bipartite graph between $1,000$ keywords and $10,475$ advertisers. Every entry in the original Yahoo-Ad dataset is a 4-tuple, which represents a "keyword-id" bid by "advertiser-id" at "time-stamp" with "price". We extract advertiser-ids and keyword-ids as nodes, and add an edge if the advertiser bids the keyword at least once. Each edge shows the "who is interested in what" relationship. This dataset will contain $11,475$ nodes and $52,567$ edges. The motivation of this experiment is to select a set of keywords that is maximally associated to advertisers, which is useful for the publisher to promote keywords to advertisers. We then consider the DM network [47] with 679 nodes representing researchers and $3,374$ edges representing collaborations between them. We simulate a researcher asking others (i.e., S_A) to spread her ideas while her competitor (i.e., S_B) promotes a competing proposal. We set the parameters of our experiments as the following. For the edge weights, Yahoo-Ad uses the weighted cascade method [26], i.e. $p(s,t) = 1/deg_-(s)$, where $deg_-(s)$ is the in-degree of node s, and weights for DM are obtained by the learned edge parameters from [47]. For Bayesian regrets, we set a prior distribution of $\mu_e \sim Beta(5w_e, 5(1-w_e))$, where w_e is the true edge weight as specified above.

We model non-strategic and strategic competitors by selecting the seed set S_B

[3]https://webscope.sandbox.yahoo.com

Table 3.2: Dataset Statistics.

Network	n	m	Average Degree
DM	679	3,374	4.96
Yahoo-Ad	11,475	52,567	4.58

uniformly at random (denoted as RD) or by running the non-competitive influence maximization algorithm (denoted as IM). In our experiments, we set $|S_A| = |S_B| = 5$ for Yahoo-Ad and $|S_A| = |S_B| = 10$ for the DM dataset, and $B > A$. Since the optimal solution given the true edge probabilities cannot be derived in polynomial time, for Yahoo-Ad, we use the greedy solution as the optimal baseline, which is a $(1 - 1/e, 1)$-approximate solution. For the DM dataset, we use the IMM solution as the optimal baseline, which is a $(1 - 1/e - \epsilon, 1 - n^{-l})$-approximate solution. For frequentist regrets, we repeat each experiment 50 times and show the average regret with 95% confidence interval. For Bayesian regrets, we draw 5 problem instances according to the prior distributions, conduct 10 experiments in each instance and report the average Bayesian regret over the 50 experiments.

Algorithms for comparison. For OCIM-TS, since the true prior distribution is unknown for the frequentist setting, we use the uninformative prior $Beta(1, 1)$ for each μ_e. For OCIM-OFU, we shrink its confidence interval by α_ρ, i.e., $\rho_i \leftarrow \alpha_\rho \sqrt{3 \ln t / 2T_i}$, to speed up the learning. The role of α_ρ represents a tradeoff between theoretical guarantees and real-world performance. $\alpha_\rho \geq 1$ provides theoretical regret bounds for the worst-case (i.e., our algorithms have sublinear regret for any problem instance) and most of the bandit literature gives regret analysis under this condition. However, in practice, we often do not face the worst problem instance. Taking a more aggressive α_ρ helps speed up the learning empirically [48], though the algorithms may incur linear regrets for bad problem instances (which are likely rare in practice), preventing us from achieving worst-case theoretical regret bounds. We compare OCIM-OFU/OCIM-TS to the ϵ-Greedy algorithm with parameter $\epsilon = 0$ (denoted as the EMP algorithm) and $\epsilon = 0.01$, which inputs the empirical mean into the offline oracle with $1 - \epsilon$ probability and otherwise selects S_A uniformly at random.

Running time. We show the average running times for different algorithms in Table 3.3. For the Yahoo-Ad dataset, OCIM-ETC is the fastest one as it only needs to call the oracle for one time before the exploitation phase. The running time of

Table 3.3: Average Running Time (second/round).

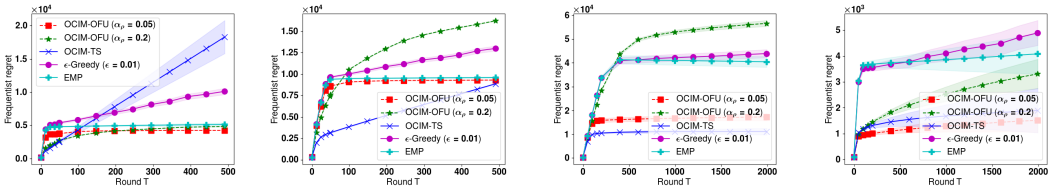
Dataset	OCIM-OFU	OCIM-TS	OCIM-ETC	ϵ-greedy	EMP
Yahoo-Ad	1.221	1.641	0.729	1.244	1.226
DM	1.142	1.195	0.621	1.173	1.125

OCIM-TS is slower than that of OCIM-OFU because it requires an extra sampling procedure to generate Thompson samples. For the DM dataset, all algorithms consume less time since the graph is smaller, but the relative order for different algorithms are preserved.

Experiments for frequentist regrets. Figures 3.3a and 3.3b show the results for Yahoo-Ad. First, the regret of OCIM-OFU grows sub-linearly with respect to round T for all α_ρ, consistent with Theorem 3.7's regret bound. Second, we can observe that OCIM-OFU is superior to EMP and ϵ-Greedy when $\alpha_\rho = 0.05$. When $\alpha_\rho = 0.2$, OCIM-OFU may have larger regret due to too much exploration. The OCIM-TS algorithm has larger slope in regrets compared to other algorithms. We speculate that such large slope comes from the uninformative prior, which requires more rounds to compensate for the mismatch of the uninformative and the true priors.

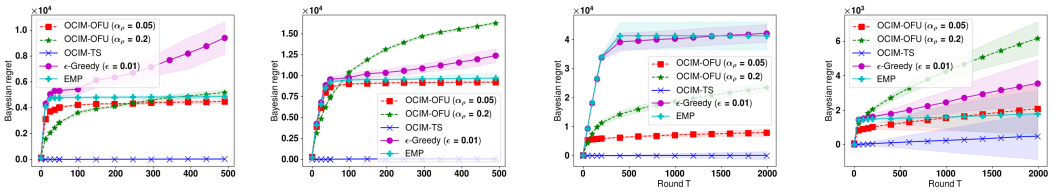
The results on the DM dataset are shown in Figs. 3.3c and 3.3d. Generally, they are consistent with those on the Yahoo-Ad dataset: OCIM-OFU also grows sub-linearly w.r.t round T. When $\alpha_\rho = 0.05$, OCIM-OFU has smaller regret than all baselines. Moreover, the difference between OCIM-OFU and the baselines for the non-strategic competitor (RD) is more significant than that of the strategic competitor's (IM), because the non-strategic competitor is less "dominant" and OCIM-OFU can carefully trade off exploration and exploitation to maximize A's influence. OCIM-TS learns faster and achieves better performance in this dataset compared to that in the Yahoo-Ad dataset.

Experiments for Bayesian regrets. We show Bayesian regrets of all algorithms in Figure 3.4. All algorithms except for OCIM-TS have similar curves. OCIM-TS, however, achieves at least two orders of magnitudes lower regret ($BayesReg(T) \approx 100$) compared with other algorithms. The reason is that OCIM-TS leverages its prior knowledge to quickly converge to the optimal solution, but other algorithms cannot use this knowledge effectively.



(a) Yahoo-Ad, RD **(b)** Yahoo-Ad, IM **(c)** DM, RD **(d)** DM, IM

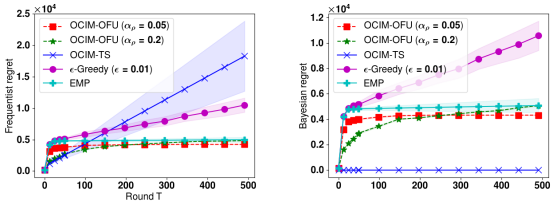
Figure 3.3: Frequentist regrets of algorithms.



(a) Yahoo-Ad, RD **(b)** Yahoo-Ad, IM **(c)** DM, RD **(d)** DM, IM

Figure 3.4: Bayesian regrets of algorithms.

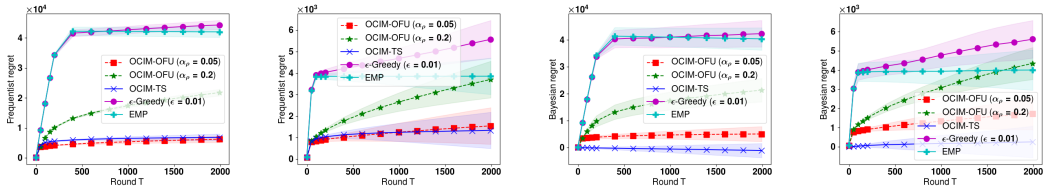
Experiments for $A > B$ Tie-breaking Rule. When we consider $A > B$ in bipartite graphs, we can trivially ignore S_B to choose S_A since the influence spread ends in one diffusion round, and OCIM becomes the online influence maximization problem without competition. We show such results in Figure 3.5. Note that the distribution of B no longer affects the performance of A when $A > B$ and we only use one figure for the IM and RD distribution. For general graphs, we use the same DM dataset and parameter settings described in Sec. 3.4.7, and the only difference is that A now dominates B. We show the results in Figure 3.6. Overall, the results and the analysis for $A > B$ are consistent with $B > A$.



(a) Yahoo-Ad, Freq. **(b)** Yahoo-Ad, Bayesian

Figure 3.5: Frequentist/Bayesian regrets for the Yahoo-Ad graph when $A > B$.

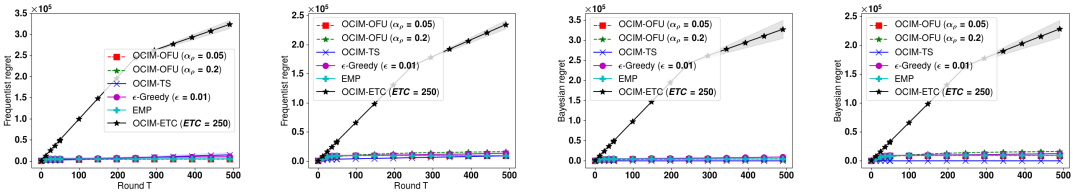
Experiments for OCIM-ETC. We show the frequentist/Bayesian regret results for the OCIM-ETC algorithm in Figure 3.7, Figure 3.8 and Figure 3.9. In Figure 3.7,



(a) DM, RD, Freq. **(b)** DM, IM, Freq. **(c)** DM, RD, Bayesian **(d)** DM, IM, Bayesian

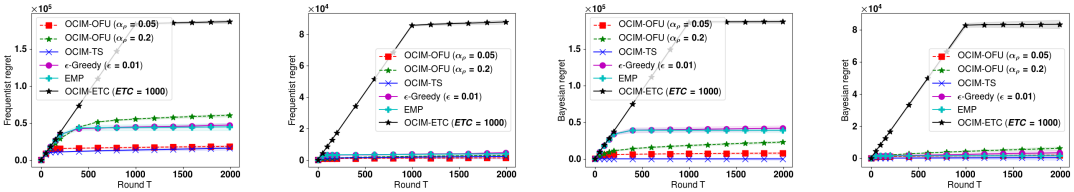
Figure 3.6: Frequentist/Bayesian regrets for the general graph DM when $A > B$.

we set exploration phase to be 250 rounds and the experiments show that we suffer linear regrets in both the exploration and the exploitation phase, meaning that the unknown parameters are under-explored. Thus we reset exploration to be 1500 and Figure 3.9 shows that OCIM-ETC now has constant regret in the exploitation phase. For DM dataset, since the node number and the edge number are less than Yahoo-Ad, we can see constant regrets after 1000 rounds of exploration in Figure 3.8. Compared with OCIM-OFU/OCIM-TS, OCIM-ETC requires more rounds to learn the unknown influence probabilities and has larger regrets than OCIM-OFU/OCIM-TS, but with sufficient exploration (which is much less than the theoretical requirements $N = (\tilde{C}m)^{\frac{2}{3}}(nk)^{-\frac{1}{3}}T^{\frac{2}{3}}(\ln T)^{\frac{1}{3}}$ in Theorem 3.9) OCIM-ETC can yield constant regrets during the exploitation phase in our experiments.



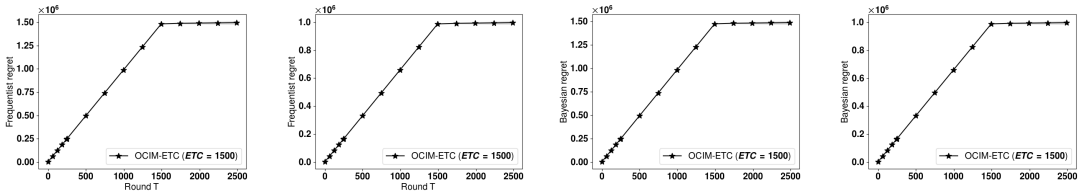
(a) RD, Frequentist **(b)** IM, Frequentist **(c)** RD, Bayesian **(d)** IM, Bayesian

Figure 3.7: Frequentist/Bayesian regrets of OCIM-ETC for the Yahoo-Ad graph.



(a) RD, Frequentist **(b)** IM, Frequentist **(c)** RD, Bayesian **(d)** IM, Bayesian

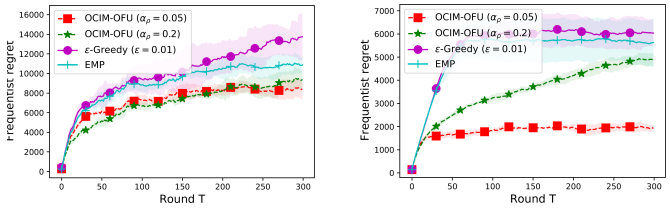
Figure 3.8: Frequentist/Bayesian regrets of OCIM-ETC for the DM graph.



(a) RD, Frequentist **(b)** IM, Frequentist **(c)** RD, Bayesian **(d)** IM, Bayesian

Figure 3.9: Frequentist/Bayesian regrets of OCIM-ETC for the Yahoo-Ad graph with 1500 rounds of exploration.

Experiments for Probabilistic Seed Distribution. For the settings where the competitor has unknown fixed seed distribution, we first run the non-competitive influence maximization algorithm for S_B. and get the best 5 seeds on Yahoo-Ad and the best 10 seeds on DM, respectively, We then consider the seed distribution of S_B as choosing each node from the best seeds with probability 0.5, i.e., the probability that choosing all best 5 seeds on Yahoo-Ad is 0.5^5 and the probability that choosing all best 10 seeds on DM is 0.5^{10}. This seed distribution of S_B is unknown to our algorithms. In our experiments, we set $|S_A| = 5$ for Yahoo-Ad and $|S_A| = 10$ for DM, and assume $B > A$. Figure 3.10 shows that OCIM-OFU is still superior to EMP and ϵ-Greedy for this setting with more complex competitor actions. We omit the results of OCIM-TS here as it requires the prior knowledge of the competitor's seed distribution. However, as long as the given prior does not differ much from the true prior, OCIM-TS will also achieve good regret results.



(a) Yahoo-Ad **(b)** DM

Figure 3.10: Frequentist regrets with unknown fixed competitor's seed distribution.

3.5 Summary

In this chapter, we study the competitive CMAB problem from the follower's perspective. We first formulate it as a general C^2MAB-T problem, then introduce four bandit algorithms for different settings with different regret guarantees. We also provide an in-depth study of its application to the online competitive influence maximization problem.

3.6 Proof

3.6.1 Proof of Theorem 3.5

Proof. Let $r_S^v(\boldsymbol{\mu})$ be the probability that node v is activated by A. From the proof of Lemma 2 in [3], we know that if for every node v and every $\boldsymbol{\mu}$ and $\boldsymbol{\mu'}$ vectors we have

$$|r_S^v(\boldsymbol{\mu}) - r_S^v(\boldsymbol{\mu'})| \leq \sum_{e \in E} p_e^S(\boldsymbol{\mu}) \, |\mu_e - \mu'_e| \,, \tag{3.15}$$

then Theorem 3.5 is true. Notice that

$$r_S^v(\boldsymbol{\mu}) = \mathbb{E}_{L \sim \boldsymbol{\mu}} \left[\mathbb{1}\{v \text{ is activated by } A \text{ under } L\} \right] \tag{3.16}$$

$$r_S^v(\boldsymbol{\mu'}) = \mathbb{E}_{L' \sim \boldsymbol{\mu'}} \left[\mathbb{1}\{v \text{ is activated by } A \text{ under } L'\} \right] \tag{3.17}$$

where L and L' are two live-edge graphs sampled under $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$, respectively. As mentioned in Sec. 3.4.3, we use an edge coupling method to compute the difference between $r_S^v(\boldsymbol{\mu})$ and $r_S^v(\boldsymbol{\mu'})$. Specifically, for each edge e, suppose we independently draw a uniform random variable X_e over $[0, 1]$, let

$$\begin{aligned}
&L(e) = L'(e) = 1, && \text{if } X_e \leq \min(\mu_e, \mu'_e) \\
&L(e) = 1, L'(e) = 0, && \text{if } \mu'_e < X_e < \mu_e \\
&L(e) = 0, L'(e) = 1, && \text{if } \mu_e < X_e < \mu'_e \\
&L(e) = L'(e) = 0, && \text{if } X_e \geq \max(\mu_e, \mu'_e)
\end{aligned}$$

where $L(e)$ represents the live/blocked state of edge e in live-edge graph L. Notice that L and L' does not have the subgraph relationship. Let $\boldsymbol{X} := (X_1, \ldots, X_e)$, the difference can be written as:

$$r_S^v(\boldsymbol{\mu}) - r_S^v(\boldsymbol{\mu'}) = \mathbb{E}_{\boldsymbol{X}}[f(S, L, v) - f(S, L', v)], \tag{3.18}$$

where $f(S, L, v) := \mathbb{1}\{v \text{ is activated by } A \text{ under } L\}$. Since $f(S, L, v) - f(S, L', v)$ could be 0, 1 or -1, we will discuss these cases separately.

1) $f(S, L, v) - f(S, L', v) = 0$.

This will not contribute to the expectation.

2) $f(S, L, v) - f(S, L', v) = 1$.

This will occur only if there exists a path such that: under L, v can be activated by A via this path, while under L', v cannot be activated by A via this path. We denote this event as $\mathcal{E}_1$. We will show that $\mathcal{E}_1$ occurs only if at least one of $\mathcal{E}_1^A$ and $\mathcal{E}_1^B$ occurs.

$\mathcal{E}_1^A$: There exists a path $u \to v_1 \to \cdots \to v_d = v$ such that:

1. u is activated by A under both L and L'
2. edge (u, v_1) is live under L but not L'

$\mathcal{E}_1^B$: There exists a path $u' \to v'_1 \to \cdots \to v'_{d'} = v$ such that:

1. u' is activated by B under both L and L'
2. edge (u', v'_1) is live under L' but not L

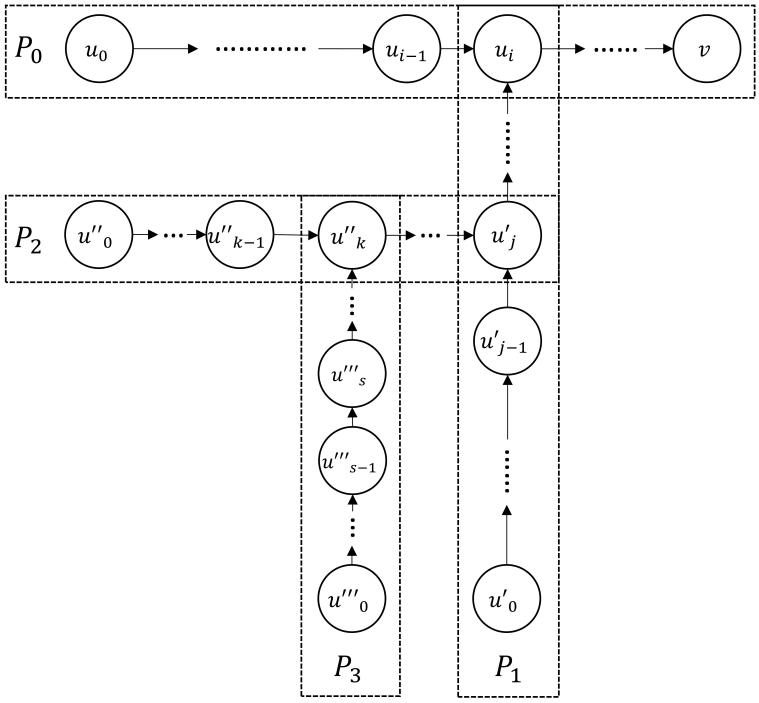


Figure 3.11: Path P_0, P_1, P_2 and P_3.

Lemma 3.1. *$\mathcal{E}_1$ occurs only if at least one of $\mathcal{E}_1^A$ and $\mathcal{E}_1^B$ occurs.*

Proof. Let us first discuss the relationship between $\mathcal{E}_1$, $\mathcal{E}_1^A$ and $\mathcal{E}_1^B$. For $\mathcal{E}_1$, if v can be activated by A under L but not L', it is because either: (a) some edge $e = (u, w)$ is live in L but blocked in L' while u is A-activated (or equivalently e is A-triggered); or (b) some edge e is live in L' but blocked in L while e is B-triggered. The former

could be relaxed to $\mathcal{E}_1^A$, and the latter could be relaxed to $\mathcal{E}_1^B$. Notice that $\mathcal{E}_1^A$ and $\mathcal{E}_1^B$ are not mutually exclusive and we are interested in the upper bound of $\mathbb{P}\{\mathcal{E}_1\}$.

Assuming $\mathcal{E}_1$ is true, consider the shortest path $P_0 := \{u_0 \to u_1 \to \cdots \to u_{l_0} = v\}$ from one seed node of A, u_0, to node v, such that under L node v is activated by A but under L' it is not. When $\mathcal{E}_1$ is true, there must exist a node that is not activated by A in P_0 under L'. We denote the first node from u_0 to v (i.e., closest to u_0) in P_0 that is not activated by A under L' as u_i.

Next, let us consider the live/blocked state of edge (u_{i-1}, u_i). We already know edge (u_{i-1}, u_i) is live under L. If edge (u_{i-1}, u_i) is blocked under L', since u_{i-1} is activated by A under both L and L', it directly becomes $\mathcal{E}_1^A$. Otherwise, if edge (u_{i-1}, u_i) is live under L', the reason that node u_i is not activated by A could only be that it is activated by B. In this case, there must exist a path $P_1 := \{u'_0 \to u'_1 \to \cdots \to u'_{l_1} = u_i\}$ from one seed node of B, u'_0, to node u_i, such that u_i is activated by B under L' but not L. This can only occur when there exists a node that is not activated by B in P_1 under L. We denote the first node from u'_0 to u'_{l_1} (i.e., closest to u'_0) in P_1 that is not activated by B under L as u'_j. Notice that when the tie-breaking rule is $A > B$, we have $l_1 < i \leq l_0$ as B should arrive at u_i earlier than A; when the tie-breaking rule is $B > A$, we have $l_1 \leq i \leq l_0$ as B should arrive at u_i no later than A. We will discuss the case of the proportional tie-breaking rule separately after the discussion of the dominance tie-breaking rules.

Then, let us consider the live/blocked state of edge (u'_{j-1}, u'_j). We already know edge (u'_{j-1}, u'_j) is live under L'. If edge (u'_{j-1}, u'_j) is blocked under L, since u'_{j-1} is activated by B under both L and L', it directly becomes $\mathcal{E}_1^B$. Otherwise, if edge (u'_{j-1}, u'_j) is live under L, the reason that node u'_j is not activated by B could only be that it is activated by A. It also means neither $\mathcal{E}_1^A$ nor $\mathcal{E}_1^B$ occurs so far. In this case, there must exist a path $P_2 := \{u''_0 \to u''_1 \to \cdots \to u''_{l_2} = u'_j\}$ from one seed node of A, u''_0, to node u'_j, such that u'_j is activated by A under L but not L'. This can only occur when there exists a node that is not activated by A in P_2 under L'. We denote the first node from u''_0 to u''_{l_2} (i.e., closest to u''_0) in P_2 that is not activated by A under L' as u''_k. Notice that when $A > B$, we have $l_2 \leq j \leq l_1 < l_0$ as A should arrive at u'_j no later than B; when $B > A$, we have $l_2 < j \leq l_1 \leq l_0$ as A should arrive at u'_j earlier than B.

Now let us consider the live/blocked state of edge (u''_{k-1}, u''_k). We already know edge (u''_{k-1}, u''_k) is live under L. If edge (u''_{k-1}, u''_k) is blocked under L', since u''_{k-1}

is activated by A under both L and L', it directly becomes $\mathcal{E}_1^A$. Otherwise, if edge (u''_{k-1}, u''_k) is live under L', the reason that node u''_k is not activated by A could only be that it is activated by B. In this case, there must exist a path $P_3 := \{u'''_0 \to u'''_1 \to \cdots \to u'''_{l_3} = u''_k\}$ from one seed node of B, u'''_0, to node u''_k, such that u''_k is activated by B under L' but not L. This can only occur when there exists a node that is not activated by B in P_3 under L. We denote the first node from u'''_0 to u'''_{l_3} (i.e., closest to u'''_0) in P_3 that is not activated by B under L as u'''_s. Notice that when $A > B$, we have $l_3 < k \leq l_2 \leq l_1$ as B should arrive at u''_k earlier than A; when $B > A$, we have $l_3 \leq k \leq l_2 < l_1$ as B should arrive at u''_k no later than A.

Again, let us consider the live/blocked state of edge (u'''_{s-1}, u'''_s). We already know edge (u'''_{s-1}, u'''_s) is live under L'. If edge (u'''_{s-1}, u'''_s) is blocked under L, since u'''_{s-1} is activated by B under both L and L', it directly becomes $\mathcal{E}_1^B$. Otherwise, if edge (u'''_{s-1}, u'''_s) is live under L, similar to the discussion above, we need to consider a new path P_4 with length l_4 and $l_4 < l_2$.

For the case of the proportional tie-breaking rule, in addition to the edge coupling, we also need to couple the permutation order [43] for each node in L and L'. More specific, for each node j, we randomly permute all of its in-neighbors, then when we need to break a tie on j, we find its activated neighbor i that is ordered first in the permutation order, and assign the state of i as j's state. Assuming the same permutation order in L and L', let us consider path P_0 and P_1 again. If $l_0 = l_1$, then u_i must be v. If $\mathcal{E}_1^A$ does not occur in P_0, then the only neighbor of v in P_1 must be ordered before the only neighbor of v in P_0 in the permutation order on v. However, if $\mathcal{E}_1^B$ does not occur in P_1, with such permutation order, it is impossible that v is activated by A under L but not L'. As a result, if neither $\mathcal{E}_1^A$ nor $\mathcal{E}_1^B$ occurs in path P_0 and P_1, we have $l_2 \leq l_1 < l_0$ in the case of the proportional tie-breaking rule.

To sum up, if neither $\mathcal{E}_1^A$ nor $\mathcal{E}_1^B$ occurs in path P_0 and P_1, we need to check whether they could occur in a new path P_2 shorter than P_0, and P_3 shorter than P_1. As a result, we only need to check whether $\mathcal{E}_1^A$ or $\mathcal{E}_1^B$ occurs in the path with only one edge. In that case, $\mathcal{E}_1^A$ or $\mathcal{E}_1^B$ occurs for sure. Thus, by induction, we conclude that at least one of $\mathcal{E}_1^A$ and $\mathcal{E}_1^B$ occurs when considering any path with more than one edge, so $\mathcal{E}_1$ will occur only if at least one of $\mathcal{E}_1^A$ and $\mathcal{E}_1^B$ occurs. □

Now, let us consider the two events in $\mathcal{E}_1^A$ for a specific edge $e = (u, v_1)$. We find that the first event $\{u$ is activated by A under both L and $L'\}$, is independent of

the second event {edge e is live under L but not L'}, since the live/blocked state of edge e does not affect the activation of its tail node u. Also, for edge $e = (u, v_1)$, the probability of these two events can be written as

$$\mathbb{P}\{u \text{ is activated by } A \text{ under } L \text{ and } L'\} = \mathbb{P}\{e \text{ is triggered by } A \text{ under } L \text{ and } L'\}, \tag{3.19}$$

$$\mathbb{P}\{e \text{ is live under } L \text{ but not } L'\} = \begin{cases} \mu_e - \mu'_e & \text{if } \mu_e > \mu'_e \\ 0 & \text{otherwise.} \end{cases} \tag{3.20}$$

As a result, we have:

$$\mathbb{P}\{\mathcal{E}_1^A\} \leq \sum_{e:\, \mu_e > \mu'_e} \mathbb{P}\{e \text{ is triggered by } A \text{ under } L \text{ and } L'\}(\mu_e - \mu'_e) \tag{3.21}$$

Since $\mathcal{E}_1^A$ and $\mathcal{E}_1^B$ are symmetric, we also have:

$$\mathbb{P}\{\mathcal{E}_1^B\} \leq \sum_{e:\, \mu'_e > \mu_e} \mathbb{P}\{e \text{ is triggered by } B \text{ under } L \text{ and } L'\}(\mu'_e - \mu_e) \tag{3.22}$$

Combining with Lemma. 3.1, we have

$$\mathbb{P}\{\mathcal{E}_1\} \leq \mathbb{P}\{\mathcal{E}_1^A\} + \mathbb{P}\{\mathcal{E}_1^B\} \tag{3.23}$$

3) $f(S, \boldsymbol{w_1}, v) - f(S, \boldsymbol{w_2}, v) = -1$.

Similar to the previous case, this will occur only if there exists a path such that: under L', v can be activated by A via this path, while under L, v cannot be activated by A via this path. We denote this event as $\mathcal{E}_{-1}$. We show that $\mathcal{E}_{-1}$ occurs only if at least one of $\mathcal{E}_{-1}^A$ and $\mathcal{E}_{-1}^B$ occurs.

$\mathcal{E}_{-1}^A$: There exists a path $u \to v_1 \to \cdots \to v_d = v$ such that:

1. u is activated by A under both L and L'
2. edge (u, v_1) is live under L' but not L

$\mathcal{E}_{-1}^B$: There exists a path $u' \to v'_1 \to \cdots \to v'_{d'} = v$ such that:

1. u' is activated by B under both L and L'
2. edge (u', v'_1) is live under L but not L'

Since they are symmetric with $\mathcal{E}_1^A$ and $\mathcal{E}_1^B$, following the same analysis, we can

get

$$\mathbb{P}\{\mathcal{E}_{-1}^{A}\} \leq \sum_{e:\,\mu'_e>\mu_e} \mathbb{P}\{e \text{ is triggered by } A \text{ under } L \text{ and } L'\}(\mu'_e - \mu_e) \tag{3.24}$$

$$\mathbb{P}\{\mathcal{E}_{-1}^{B}\} \leq \sum_{e:\,\mu_e>\mu'_e} \mathbb{P}\{e \text{ is triggered by } B \text{ under } L \text{ and } L'\}(\mu_e - \mu'_e) \tag{3.25}$$

$$\mathbb{P}\{\mathcal{E}_{-1}\} \leq \mathbb{P}\{\mathcal{E}_{-1}^{A}\} + \mathbb{P}\{\mathcal{E}_{-1}^{B}\} \tag{3.26}$$

Combining all cases together, we have:

$$\begin{aligned} |r_S^v(\boldsymbol{\mu}) - r_S^v(\boldsymbol{\mu'})| &= |\mathbb{E}_{\boldsymbol{X}}[f(S,L,v) - f(S,L',v)]| \\ &\leq |1 \cdot \mathbb{P}\{\mathcal{E}_1\} + (-1) \cdot \mathbb{P}\{\mathcal{E}_{-1}\}| \\ &\leq \left|1 \cdot \left(\mathbb{P}\{\mathcal{E}_1^{A}\} + \mathbb{P}\{\mathcal{E}_1^{B}\}\right) + (-1) \cdot \left(\mathbb{P}\{\mathcal{E}_{-1}^{A}\} + \mathbb{P}\{\mathcal{E}_{-1}^{B}\}\right)\right| \\ &\leq \sum_{e\in E} \mathbb{P}\{e \text{ is triggered by } A \text{ or } B \text{ under } L \text{ and } L'\} \left|\mu_e - \mu'_e\right|. \end{aligned} \tag{3.27}$$

The last inequality above is due to:

$$|\mathbb{P}\{\mathcal{E}_1^{A}\} - \mathbb{P}\{\mathcal{E}_{-1}^{B}\}| \leq \sum_{e:\,\mu_e>\mu'_e} \mathbb{P}\{e \text{ is triggered by } A \text{ or } B \text{ under } L \text{ and } L'\}|\mu_e - \mu'_e|$$

$$|\mathbb{P}\{\mathcal{E}_1^{B}\} - \mathbb{P}\{\mathcal{E}_{-1}^{A}\}| \leq \sum_{e:\,\mu'_e>\mu_e} \mathbb{P}\{e \text{ is triggered by } A \text{ or } B \text{ under } L \text{ and } L'\}|\mu_e - \mu'_e|$$

Notice that Eq.(3.27) could be relaxed to:

$$\begin{aligned} |r_S^v(\boldsymbol{\mu}) - r_S^v(\boldsymbol{\mu'})| &\leq \sum_{e\in E} \mathbb{P}\{e \text{ is triggered by } A \text{ or } B \text{ under } L\} \left|\mu_e - \mu'_e\right| \\ &\leq \sum_{e\in E} p_e^S(\boldsymbol{\mu}) \left|\mu_e - \mu'_e\right|. \end{aligned} \tag{3.28}$$

□

3.6.2 Proof of Theorem 3.6

Proof. We define $G^{(t)}$ as the feedback of OCIM in round t, which includes the outcomes of $X_i^{(t)}$ for all $i \in \tau_t$. We denote by $\mathcal{F}_{t-1}$ the history $(S^{(1)}, G^{(1)}, \cdots, S^{(t-1)}, G^{(t-1)})$

of observations available to the player when choosing an action $S^{(t)}$. For the Bayesian analysis, we assume the mean vector $\boldsymbol{\mu}$ follows a prior distribution $\mathcal{Q}$. In round t, given $\mathcal{F}_{t-1}$, we define the posterior distribution of $\boldsymbol{\mu}$ as $\mathcal{Q}^{(t)}$ (i.e., $\boldsymbol{\mu}^{(t)} \sim \mathcal{Q}^{(t)}$ where $\boldsymbol{\mu}^{(t)}$ is given in Alg. 3.5). As mentioned in Section 3.4.4, OCIM-TS allows any benchmark offline oracles, including approximation oracles. We consider a general benchmark oracle $\mathcal{O}(S_B, \boldsymbol{\mu})$. As oracle $\mathcal{O}$ might be a randomized policy (e.g., an (α, β)-approximation oracle with success probability β), we use a random variable $\omega \sim \Omega$ to represent all its randomness. In order to discuss the performance of OCIM-TS with oracle $\mathcal{O}$, we rewrite the Bayesian regret in Eq.(3.11) as

$$BayesReg(T) = \mathbb{E}_{\omega\sim\Omega,\boldsymbol{\mu}\sim\mathcal{Q}}\left[\sum_{t=1}^{T}\left(r_{\mathcal{O}(S_B^{(t)},\boldsymbol{\mu})}(\boldsymbol{\mu}) - r_{\mathcal{O}(S_B^{(t)},\boldsymbol{\mu_t})}(\boldsymbol{\mu})\right)\right]. \tag{3.29}$$

Notice that $\mathcal{O}(S_B^{(t)}, \boldsymbol{\mu})$ is the action taken by the player if the true $\boldsymbol{\mu}$ is known, while $\mathcal{O}(S_B^{(t)}, \boldsymbol{\mu}_t)$ is the real action chosen by OCIM-TS. The original regret definition in Eq.(3.11) is a special case of Eq.(3.29) for an (α, β)-approximation oracle, and will focus on this general form in this proof.

The key step to derive the Bayesian regret bound of OCIM-TS is to show that the conditional distributions of $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_t$ given $\mathcal{F}_{t-1}$ are the same:

$$\mathbb{P}(\boldsymbol{\mu} = \cdot \mid \mathcal{F}_{t-1}) = \mathbb{P}(\boldsymbol{\mu}_t = \cdot \mid \mathcal{F}_{t-1}), \tag{3.30}$$

which is true since we use Thompson sampling to update the posterior distribution

of $\boldsymbol{\mu}$. With this finding, we consider the Bayesian regret in Eq.(3.11):

$$
\begin{aligned}
&BayesReg(T)\\
=&\mathbb{E}_{\omega\sim\Omega}\left[\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{\mu}\sim\mathcal{Q},\boldsymbol{\mu}_t\sim\mathcal{Q}_t}\left[r_{\mathcal{O}(S_B^{(t)},\boldsymbol{\mu})}(\boldsymbol{\mu})-r_{\mathcal{O}(S_B^{(t)},\boldsymbol{\mu}_t)}(\boldsymbol{\mu})\right]\right] &(3.31)\\
=&\mathbb{E}_{\omega\sim\Omega}\left[\sum_{t=1}^{T}\mathbb{E}_{\mathcal{F}_{t-1}}\left[\mathbb{E}_{\boldsymbol{\mu}\sim\mathcal{Q},\boldsymbol{\mu}_t\sim\mathcal{Q}_t}\left[r_{\mathcal{O}(S_B^{(t)},\boldsymbol{\mu})}(\boldsymbol{\mu})-r_{\mathcal{O}(S_B^{(t)},\boldsymbol{\mu}_t)}(\boldsymbol{\mu})\right]\mid\mathcal{F}_{t-1}\right]\right] &(3.32)\\
=&\mathbb{E}_{\omega\sim\Omega}\left[\sum_{t=1}^{T}\mathbb{E}_{\mathcal{F}_{t-1}}\left[\mathbb{E}_{\boldsymbol{\mu}\sim\mathcal{Q},\boldsymbol{\mu}_t\sim\mathcal{Q}_t}\left[r_{\mathcal{O}(S_B^{(t)},\boldsymbol{\mu}_t)}(\boldsymbol{\mu}_t)-r_{\mathcal{O}(S_B^{(t)},\boldsymbol{\mu}_t)}(\boldsymbol{\mu})\right]\mid\mathcal{F}_{t-1}\right]\right] &(3.33)\\
=&\mathbb{E}\left[\sum_{t=1}^{T}\left[r_{\mathcal{O}(S_B^{(t)},\boldsymbol{\mu}_t)}(\boldsymbol{\mu}_t)-r_{\mathcal{O}(S_B^{(t)},\boldsymbol{\mu}_t)}(\boldsymbol{\mu})\right]\right], &(3.34)
\end{aligned}
$$

where Eq.(3.33) comes from applying Eq.(3.30) to Eq.(3.32). Let $S_t=\mathcal{O}(S_B^{(t)},\boldsymbol{\mu}_t)$ and $\mathcal{C}_t=\{\boldsymbol{\mu}':|\mu_i'-\hat{\mu}_{i,t}|\le\rho_{i,t},\forall i\}$, where $\rho_{i,t}=\sqrt{3\ln t/2T_{i,t-1}}$ and $T_{i,t-1}$ is the total number of times arm i is played until round t. We define $\Delta_{S_t}=r_{S_t}(\boldsymbol{\mu}_t)-r_{S_t}(\boldsymbol{\mu})$ and $M=\sqrt{576\tilde{C}^2mK\ln T/T}$. By Eq.(3.34), we have

$$
\begin{aligned}
&BayesReg(T)\\
=&\mathbb{E}[\sum_{t=1}^{T}\Delta_{S_t}] &(3.35)\\
\le&\underbrace{\mathbb{E}\left[\sum_{t=1}^{T}\Delta_{S_t}\mathbb{I}\{\Delta_{S_t}\ge M,\boldsymbol{\mu}_t\in\mathcal{C}_t,\boldsymbol{\mu}\in\mathcal{C}_t,\mathcal{N}_t^{\mathrm{t}}\}\right]}_{(a)}+\underbrace{\mathbb{E}[\sum_{t=1}^{T}\Delta_{S_t}\mathbb{I}\{\boldsymbol{\mu}_t\notin\mathcal{C}_t\}]+\mathbb{E}[\sum_{t=1}^{T}\Delta_{S_t}\mathbb{I}\{\boldsymbol{\mu}\notin\mathcal{C}_t\}]}_{(b)}\\
&+\underbrace{\mathbb{E}[\sum_{t=1}^{T}\Delta_{S_t}\mathbb{I}\{\Delta_{S_t}\le M\}]}_{(c)}+\underbrace{\mathbb{E}[\sum_{t=1}^{T}\Delta_{S_t}\mathbb{I}\{\neg\mathcal{N}_t^{\mathrm{t}}\}]}_{(d)} &(3.36)
\end{aligned}
$$

We can bound these three terms separately. For term (a), when $\boldsymbol{\mu}_t\in\mathcal{C}_t,\boldsymbol{\mu}\in\mathcal{C}_t$, we could bound $|\mu_{i,t}-\mu_i|\le|\mu_{i,t}-\hat{\mu}_{i,t}|+|\mu_i-\hat{\mu}_{i,t}|\le 2\rho_{i,t},\forall i$. When $\Delta_{S_t}\ge M$ and $\mathcal{N}_t^{\mathrm{t}}$ (Definition 7 in [3]) holds, by the proof of Lemma 5 in [3], we have $\Delta_{S_t}\le\sum_{i\in\tilde{S}_t}\kappa_{j_i,T}(M_i,N_{i,j_i,t-1})$ where $\tilde{S}_t$ is the set of arms triggered by S_t and $\kappa_{j_i,T}(M_i,N_{i,j_i,t-1})$ is defined in [3]. We have

$$
\begin{aligned}
(a) &= \mathbb{E}\left[\sum_{t=1}^{T} \Delta_{S_t} \mathbb{I}\{\Delta_{S_t} \geq M, \boldsymbol{\mu}_t \in \mathcal{C}_t, \boldsymbol{\mu} \in \mathcal{C}_t, \mathcal{N}_t^{\mathrm{t}}\}\right] \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T} \sum_{i \in \tilde{S}_t} \kappa_{j_i,T}(M_i, N_{i,j_i,t-1})\right] \\
&\leq \mathbb{E}\left[\sum_{i \in [m]} \sum_{j=1}^{+\infty} \sum_{s=0}^{N_{i,j,T}-1} \kappa_{j,T}(M, s)\right] \\
&\leq 4\tilde{C}m + \sum_{i \in [m]} \frac{576\tilde{C}^2 K \ln T}{M}
\end{aligned}
$$

For term (b), we can observe that $\mathbb{E}[\mathbb{I}\{\boldsymbol{\mu} \in \mathcal{C}_t\}|\mathcal{F}_{t-1}] = \mathbb{E}[\mathbb{I}\{\boldsymbol{\mu}_t \in \mathcal{C}_t\}|\mathcal{F}_{t-1}]$, since $\mathcal{C}_t$ is determined given $\mathcal{F}_{t-1}$, and given $\mathcal{F}_{t-1}$, $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_t$ follow the same distribution. Since $\max_{S_t} \Delta_{S_t} \leq n$, we have

$$
\begin{aligned}
(b) &= \mathbb{E}[\sum_{t=1}^{T} \Delta_{S_t} \mathbb{I}\{\boldsymbol{\mu}_t \notin \mathcal{C}_t\}] + \mathbb{E}[\sum_{t=1}^{T} \Delta_{S_t} \mathbb{I}\{\boldsymbol{\mu} \notin \mathcal{C}_t\}] \\
&\leq n\left(\mathbb{E}[\sum_{t=1}^{T} \mathbb{I}\{\boldsymbol{\mu}_t \notin \mathcal{C}_t\}] + \mathbb{E}[\sum_{t=1}^{T} \mathbb{I}\{\boldsymbol{\mu} \notin \mathcal{C}_t\}]\right) \\
&= n\left(\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}\left[\mathbb{I}\{\boldsymbol{\mu}_t \notin \mathcal{C}_t\}|\mathcal{F}_{t-1}\right]\right]\right) + n\left(\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}\left[\mathbb{I}\{\boldsymbol{\mu} \notin \mathcal{C}_t\}|\mathcal{F}_{t-1}\right]\right]\right) \\
&= 2n\left(\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}\left[\mathbb{I}\{\boldsymbol{\mu} \notin \mathcal{C}_t\}|\mathcal{F}_{t-1}\right]\right]\right) \\
&= 2n\left(\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}\{\boldsymbol{\mu} \notin \mathcal{C}_t\}\right]\right) \\
&= 2n\left(\sum_{t=1}^{T} \mathbb{P}\left(\boldsymbol{\mu} \notin \mathcal{C}_t\right)\right) \\
&\leq \frac{2\pi^2 mn}{3}
\end{aligned}
$$

For term (c), we can bound it by

$$(c) = \mathbb{E}[\sum_{t=1}^{T} \Delta_{S_t} \mathbb{I}\{\Delta_{S_t} \leq M\}] \leq TM$$

For term (d), similar to Eq.(20) in [3], we have

$$(d) = \mathbb{E}[\sum_{t=1}^{T} \Delta_{S_t} \mathbb{I}\{\neg \mathcal{N}_t^{\mathrm{t}}\}] \leq \frac{\pi^2}{6} \cdot \sum_{i \in [m]} j_{\max}^i \cdot n$$

Combine them together, we have

$$BayesReg(T) \leq 4\tilde{C}m + \sum_{i \in [m]} \frac{576\tilde{C}^2 K \ln T}{M} + \frac{2\pi^2 mn}{3} + TM + \frac{\pi^2}{6} \cdot \sum_{i \in [m]} j_{\max}(M) \cdot n$$

where $j_{\max}(M) = \left\lceil \log_2 \frac{2\tilde{C}K}{M} \right\rceil_0$. Take $M = \sqrt{576\tilde{C}^2 mK \ln T / T}$, we finally get finally get the Bayesian regret bound of TS-OCIM:

$$BayesReg(T) \leq 12\tilde{C}\sqrt{mKT \ln T} + 2\tilde{C}m + \left(\left\lceil \log_2 \frac{T}{18 \ln T} \right\rceil_0 + 4 \right) \cdot \frac{\pi^2}{6} \cdot n \cdot m.$$

□

3.6.3 Proof of Theorem 3.7

Proof. We first introduce the following definitions to assist our analysis. Recall that $\boldsymbol{\mathcal{S}}^{(t)}$ is the action space in round t. We define the reward gap $\Delta_S^{(t)} = \max(0, \alpha \cdot \mathrm{opt}^{(t)}(\boldsymbol{\mu}) - r_S(\boldsymbol{\mu}))$ for all actions $S \in \boldsymbol{\mathcal{S}}^{(t)}$. For each base arm i, we define $\Delta_{\max}^{i,T} = \max_{t \in [T]} \sup_{S \in \boldsymbol{\mathcal{S}}^{(t)} : p_i^S(\boldsymbol{\mu}) > 0, \Delta_S^{(t)} > 0} \Delta_S^{(t)}$ and $\Delta_{\min}^{i,T} = \min_{t \in [T]} \inf_{S \in \mathcal{S}^{(t)} : p_i^S(\boldsymbol{\mu}) > 0, \Delta_S^{(t)} > 0} \Delta_S^{(t)}$. If there is no action S such that $p_i^S(\boldsymbol{\mu}) > 0$ and $\Delta_S^{(t)} > 0$, we define $\Delta_{\max}^{i,T} = 0$ and $\Delta_{\min}^{i,T} = +\infty$. We define $\Delta_{\max}^{(T)} = \max_{i \in [m]} \Delta_{\max}^{i,T}$ and $\Delta_{\min}^{(T)} = \min_{i \in [m]} \Delta_{\min}^{i,T}$. Let $\widetilde{S} = \{i \in [m] \mid p_i^S(\boldsymbol{\mu}) > 0\}$ be the set of arms that can be triggered by S. We define $K = \max_{S \in \boldsymbol{\mathcal{S}}^{(t)}} |\widetilde{S}|$ as the largest number of arms could be triggered by a feasible action. We use $\lceil x \rceil_0$ to denote $\max\{\lceil x \rceil, 0\}$. If $\Delta_{\min}^{(T)} > 0$, we provide the

distribution-dependent bound of the OCIM-OFU algorithm.

$$\mathrm{Reg}_{\alpha,\beta}(T;\boldsymbol{\mu}) \leq \sum_{i\in[m]} \frac{576\tilde{C}^2 K \ln T}{\Delta_{\min}^{i,T}} + 4\tilde{C}m + \sum_{i\in[m]} \left(\left\lceil \log_2 \frac{2\tilde{C}K}{\Delta_{\min}^{i,T}} \right\rceil_0 + 2 \right) \cdot \frac{\pi^2}{6} \cdot \Delta_{\max}^{(T)}.$$

To prove the distribution-dependent and the distribution-independent regret bounds, we generally follow the proof of Theorem 1 in [3]. However, since we extend the original CMAB problem to a new contextual setting where the action space $\boldsymbol{\mathcal{S}}^{(t)}$ is the context, and monotonicity does not hold in the OCIM setting, we need to modify their analysis to tackle these changes. We introduce a positive real number M_i for each arm i and define $M_{S^{(t)}} = \max_{i\in\tilde{S}^{(t)}} M_i$. Define

$$\kappa_{j,T}(M,s) = \begin{cases} 4\cdot 2^{-j}\tilde{C}, & \text{if } s = 0, \\ 2\tilde{C}\sqrt{\frac{72\cdot 2^{-j}\ln T}{s}}, & \text{if } 1 \leq s \leq \ell_{j,T}(M), \\ 0, & \text{if } s \geq \ell_{j,T}(M)+1, \end{cases}$$

where

$$\ell_{j,T}(M) = \left\lfloor \frac{288\cdot 2^{-j}\tilde{C}^2 K^2 \ln T}{M^2} \right\rfloor.$$

Let $\mathcal{N}_t^{\mathrm{s}}$ be the event that at the beginning of round t, for every arm $i \in [m]$, $|\hat{\mu}_{i,t} - \mu_i| \leq 2\rho_{i,t}$. Let $\mathcal{H}_t$ be the event that at round t oracle $\widetilde{\mathcal{O}}$ outputs a solution, $S^{(t)} = \{S_A^{(t)}, S_B^{(t)}\}$ and $\boldsymbol{\mu}^{(t)} = (\mu_1^{(t)}, \ldots, \mu_m^{(t)})$, such that $r_{S^{(t)}}(\boldsymbol{\mu}^{(t)}) < \alpha \cdot r_{S^*}(\boldsymbol{\mu}^*)$, i.e., oracle $\widetilde{\mathcal{O}}$ fails to output an α-approximate solution. Let $\mathcal{N}_t^{\mathrm{t}}$ be the event that the triggering is nice at the beginning of round t (Definition 7 in [3]). The following lemma explains how κ contributes to the regret.

Lemma 3.2. *For any vector $\{M_i\}_{i\in[m]}$ of positive real numbers and $1 \leq t \leq T$, if $\{\Delta_{S^{(t)}}^{(t)} \geq M_{S^{(t)}}\}, \neg\mathcal{H}_t, \mathcal{N}_t^s$ and $\mathcal{N}_t^t$ hold, we have*

$$\Delta_{S^{(t)}}^{(t)} \leq \sum_{i\in\tilde{S}^{(t)}} \kappa_{j_i,T}(M_i, N_{i,j_i,t-1}),$$

where j_i is the index of the TP group with $S^{(t)} \in \mathcal{S}_{i,j_i}$ (see Definition 5 in [3]).

Proof. By $\mathcal{N}_t^{\mathrm{s}}$ and $0 \leq \mu_i \leq 1$ for all $i \in [m]$, we have

$$\forall i \in [m], \mu_i \in c_{i,t} = \left[(\hat{\mu}_{i,t} - \rho_{i,t})^{0+}, (\hat{\mu}_{i,t} + \rho_{i,t})^{1-} \right]. \tag{3.37}$$

It means that we have the correct estimated range of μ_i for all $i \in [m]$ at round t. Combining with $\neg\mathcal{H}_t$ for the offline oracle $\widetilde{\mathcal{O}}$, we have

$$r_{S^{(t)}}(\boldsymbol{\mu}^{(t)}) \geq \alpha \cdot r_{S^*}(\boldsymbol{\mu}^*) \geq \alpha \cdot \mathrm{opt}^{(t)}(\boldsymbol{\mu}) = r_{S^{(t)}}(\boldsymbol{\mu}) + \Delta^{(t)}_{S^{(t)}}. \tag{3.38}$$

By the TPM condition in Theorem. 3.5, we have

$$\Delta^{(t)}_{S^{(t)}} \leq r_{S^{(t)}}(\boldsymbol{\mu}^{(t)}) - r_{S^{(t)}}(\boldsymbol{\mu}) \leq \tilde{C} \sum_{i\in[m]} p_i^{S^{(t)}}(\boldsymbol{\mu})|\mu_i^{(t)} - \mu_i|. \tag{3.39}$$

We want to bound $\Delta^{(t)}_{S^{(t)}}$ by bounding $p_i^{S^{(t)}}(\boldsymbol{\mu})|\mu_i^{(t)} - \mu_i|$. We first perform a transformation. Since $\Delta^{(t)}_{S^{(t)}} \geq M_{S^{(t)}}$, we have $\tilde{C}\sum_{i\in[m]} p_i^{S^{(t)}}(\boldsymbol{\mu})|\mu_i^{(t)} - \mu_i| \geq \Delta^{(t)}_{S^{(t)}} \geq M_{S^{(t)}}$. Then we have

$$\begin{aligned}
\Delta^{(t)}_{S^{(t)}} &\leq \tilde{C} \sum_{i\in[m]} p_i^{S^{(t)}}(\boldsymbol{\mu})|\mu_i^{(t)} - \mu_i| \\
&\leq -M_{S^{(t)}} + 2\tilde{C} \sum_{i\in[m]} p_i^{S^{(t)}}(\boldsymbol{\mu})|\mu_i^{(t)} - \mu_i| \\
&\leq 2\tilde{C} \sum_{i\in[m]} \left[p_i^{S^{(t)}}(\boldsymbol{\mu})|\mu_i^{(t)} - \mu_i| - \frac{M_i}{2\tilde{C}K} \right].
\end{aligned} \tag{3.40}$$

In fact, if $\mathcal{N}_t^{\mathrm{s}}$ holds and $\mu_i^{(t)} \in c_{i,t}$ for all $i \in [m]$,

$$\forall i \in [m], |\mu_i^{(t)} - \mu_i| \leq 2\rho_{i,t} = 2\sqrt{\frac{3\ln t}{2T_{i,t-1}}}. \tag{3.41}$$

So far, all requirements on bounding Δ_{S_t} in Lemma 5 from [3] are also satisfied by $\Delta^{(t)}_{S^{(t)}}$ of OCIM-OFU algorithm in the OCIM setting without monotonicity. We can then follow the same steps to bound $p_i^{S^{(t)}}(\boldsymbol{\mu})|\mu_i^{(t)} - \mu_i|$ in the two cases they considered (combining their Eq.(11)-(13)) and get

$$\begin{aligned}
\Delta^{(t)}_{S^{(t)}} &\leq 2\tilde{C} \sum_{i\in[m]} \left[p_i^{S^{(t)}}(\boldsymbol{\mu})|\mu_i^{(t)} - \mu_i| - \frac{M_i}{2\tilde{C}K} \right] \\
&\leq \sum_{i\in\tilde{S}^{(t)}} \kappa_{j_i,T}(M_i, N_{i,j_i,t-1}).
\end{aligned}$$

□

With Lemma 3.2, we can follow the proof of Lemma 6 in [3] to bound the regret when $\{\Delta^{(t)}_{S^{(t)}} \geq M_{S^{(t)}}\}, \neg\mathcal{H}_t, \mathcal{N}^{\mathrm{s}}_t$ and $\mathcal{N}^{\mathrm{t}}_t$ hold.

$$Reg(\{\Delta^{(t)}_{S^{(t)}} \geq M_{S^{(t)}}\} \wedge \neg\mathcal{H}_t \wedge \mathcal{N}^{\mathrm{s}}_t \wedge \mathcal{N}^{\mathrm{t}}_t) \leq \sum_{i\in[m]} \frac{576\tilde{C}^2 K \ln T}{M_i} + 4\tilde{C}m. \tag{3.42}$$

Finally, we take $M_i = \Delta^{i,T}_{\min}$. If $\Delta^{(t)}_{S^{(t)}} < M_{S^{(t)}}$, then $\Delta^{(t)}_{S^{(t)}} = 0$, since we have either $\tilde{S}^{(t)} = \emptyset$ or $\Delta^{(t)}_{S^{(t)}} < M_{S^{(t)}} \leq M_i$ for some $i \in \tilde{S}^{(t)}$. Thus, no regret is accumulated when $\Delta^{(t)}_{S^{(t)}} < M_{S^{(t)}}$. Following Eq.(17)-(21) in [3], we can derive the distribution-dependent regret bound

$$\mathrm{Reg}_{\alpha,\beta}(T;\boldsymbol{\mu}) \leq \textstyle\sum_{i\in[m]} \frac{576\tilde{C}^2 K \ln T}{\Delta^{i,T}_{\min}} + 4\tilde{C}m + \sum_{i\in[m]} \left(\left\lceil \log_2 \frac{2\tilde{C}K}{\Delta^{i,T}_{\min}} \right\rceil_0 + 2\right) \cdot \frac{\pi^2}{6} \cdot \Delta^{(T)}_{\max}. \tag{3.43}$$

To derive the distribution-independent bound, we take $M_i = M = \sqrt{(576\tilde{C}^2 mK \ln T)/T}$, follow Eq.(23) in [3] and get

$$\mathrm{Reg}_{\alpha,\beta}(T;\boldsymbol{\mu}) \leq 12\tilde{C}\sqrt{mKT\ln T} + 2\tilde{C}m + \left(\left\lceil \log_2 \tfrac{T}{18\ln T} \right\rceil_0 + 2\right) \cdot \tfrac{\pi^2}{6} \cdot n \cdot m. \tag{3.44}$$

□

3.6.4 Computational Efficiency of OCIM-OFU

Proof of Theorem 3.8

Proof. In order to prove Theorem 3.8, we first introduce a new optimization problem denoted as P_1: given S, the new problem aims to find the optimal μ_i for one edge i to maximize $r_S(\boldsymbol{\mu})$, while fixing the values of all others. The following lemma shows it is #P-hard.

Lemma 3.3. *Given S and fixing μ_e for all $e \neq i$, finding the optimal $\mu_i \in c_i$ for one edge i that maximizes $r_S(\boldsymbol{\mu})$ is #P-hard.*

Proof. We prove the hardness of this optimization problem via a reduction from the influence computation problem. We first consider a general graph G_0 with n nodes and m edges, where all influence probabilities on edges are set to 1/2. Given

S_A, computing the influence spread of A in such a graph is #P-hard. Notice that there is no seed set of B in G_0. Now let us take one node v in G_0 and denote its activation probability by A as $h_A(G_0, S_A, v)$. Actually, computing $h_A(G_0, S_A, v)$ is also #P-hard and we want to show that it can be reduced to our optimization problem in polynomial time.

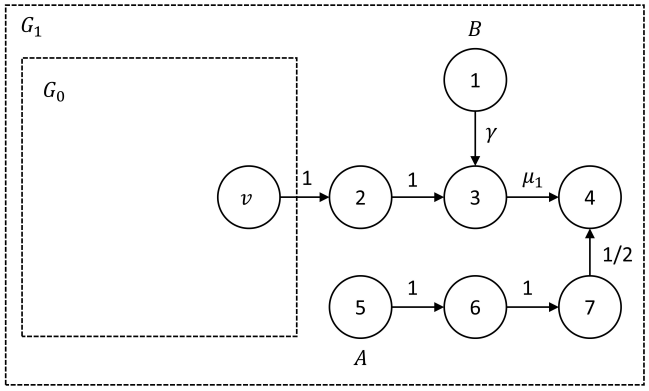


Figure 3.12: Construction of G_1 based on G_0.

We first construct a new graph G_1 based on G_0. For G_1, we keep G_0 and S_A unchanged, then add several nodes and edges as shown in Fig. 3.12. We add node 1 to the seed set of B and node 5 to the seed set of A, so the joint action $S = \{S_A \cup \{5\}, S_B = \{1\}\}$. In this new graph G_1, we consider the optimization problem of finding the optimal μ_1 (influence probability on edge $(3, 4)$) within its range c_1 that maximizes $r_S(\boldsymbol{\mu})$. Notice that the influence probability γ on edge $(1, 3)$ is a constant and μ_1 would only affect the activation probability of node 4. We denote the activation probability by A of node 4 as $h_A(G_1, S, 4)$. In order to maximize $r_S(\boldsymbol{\mu})$, we only need to maximize $h_A(G_1, S, 4)$. It can be written as:

$$h_A(G_1, S, 4) = \frac{1}{2}\Big[(1-\gamma)\cdot h_A(G_1, S, v) - \gamma\Big]\cdot \mu_1 + \frac{1}{2}. \tag{3.45}$$

It is easy to see $h_A(G_1, S, 4)$ has a linear relationship with μ_1, so the optimal μ_1 could only be either the lower or upper bound of its range c_1. Assuming we can solve the optimization problem of finding the optimal μ_1, then we can determine the sign of μ_1's coefficient in Eq.(3.45): if the optimal μ_1 is the upper bound value in c_1, we have $(1-\gamma)\cdot h_A(G_1, S, v) - \gamma \geq 0$; otherwise, $(1-\gamma)\cdot h_A(G_1, S, v) - \gamma < 0$. It means

we can answer the question that whether $h_A(G_1, S, v)$ is larger (or smaller) than $\frac{\gamma}{1-\gamma}$. Notice that $h_A(G_0, S_A, v) = h_A(G_1, S, v)$, so we can manually change the value of γ to check whether $h_A(G_0, S_A, v)$ is larger (or smaller) than $x = \frac{\gamma}{1-\gamma}$ for any $x \in [0, 1]$, Recall that all edge probabilities in G_0 are set to $1/2$, so the highest precision of $h_A(G_0, S_A, v)$ should be 2^{-m}. Hence, we can use a binary search algorithm to find the exact value of $h_A(G_0, S_A, v)$ in at most m times. It means computing the activation probability of v in G_0 can be reduced to the optimization problem of finding the optimal μ_1 in G_1, which completes the proof. □

We then show that P_1 is a special case of Eq.(3.13). The main idea is to relax the constraints $|S_A| \leq k, S = \{S_A, S_B\}$ in Eq.(3.13) and show that it can find the optimal $\boldsymbol{\mu}$ for any given S. Consider a graph G with n nodes and a given seed set $S = \{S_A, S_B\}$. We construct a new graph G' by manually add additional $n + 1$ nodes pointing from each seed node in S_A. If we can solve the optimization problem Eq.(3.13) in the new graph G', since S_A must be the optimal seed set of A and the added nodes will not affect the prorogation in G, we will also find the optimal μ_i's in the original graph G for the given S. Then, it is easy to see P_1 is a special case of Eq.(3.13) since P_1 only find the optimal μ_i for one edge i. With Lemma 3.3, we know Eq.(3.13) is also #P-hard.

□

Non-submodularity of $g(S)$

In Section 3.4.5, we introduce $g(S) = \max_{\boldsymbol{\mu}} r_S(\boldsymbol{\mu})$, which is an upper bound function of $r_S(\boldsymbol{\mu})$ for each S. If $g(S)$ is submodular over S, we can use a greedy algorithm on $g(S)$ to find an approximate solution. However, the following example in Fig. 3.13 shows that $g(S)$ is not submodular.

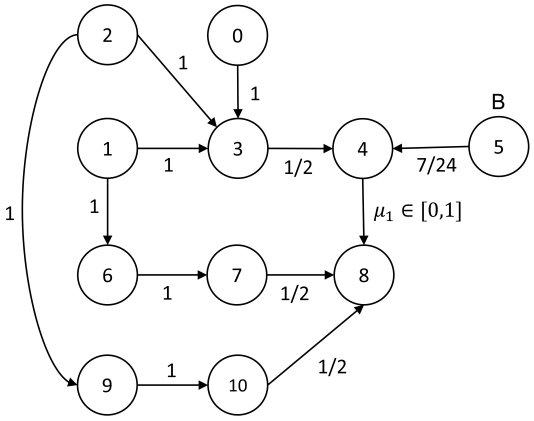


Figure 3.13: Example showing that $g(S)$ is not submodular.

In Fig. 3.13, the numbers attached to edges are influence probabilities. Only the influence probability of edge $(4, 8)$ is a variable and we denote it as μ_1. We assume $\mu_1 \in [0, 1]$ and $S_B = \{5\}$. Let us consider some choices of S_A. When S_A is chosen as $\{0\}, \{0, 1\}$ or $\{0, 2\}$, the optimal μ_1 that maximizes $r_S(\boldsymbol{\mu})$ is 1; when S_A is chosen as $\{0, 1, 2\}$, the optimal μ_1 that maximizes $r_S(\boldsymbol{\mu})$ is 0. Based on this observation, we can calculate $g(S)$ (assuming $S_B = \{5\}$):

$$
\begin{aligned}
g(S_A = \{0\}) &= 2 + \frac{17}{24}, \\
g(S_A = \{0, 1\}) &= 5 + \frac{17}{24} \times \frac{4}{5}, \\
g(S_A = \{0, 2\}) &= 5 + \frac{17}{24} \times \frac{4}{5}, \\
g(S_A = \{0, 1, 2\}) &= 8 + \frac{17}{24} \times \frac{1}{2} + \frac{3}{4}.
\end{aligned}
$$

Thus we have

$$
g(S_A = \{0, 1\}) + g(S_A = \{0, 2\}) < g(S_A = \{0\}) + g(S_A = \{0, 1, 2\}), \tag{3.46}
$$

which is contrary to submodularity.

Bipartite Graph

We consider a weighted bipartite graph $G = (L, R, E)$ where each edge (u, v) is associated with a probability $p(u, v)$. Given the competitor's seed set $S_B \subseteq L$, we need to choose k nodes from L as S_A that maximizes the expected number of nodes activated by A in R, where a node $v \in R$ can be activated by a node $u \in L$ with an independent probability of $p(u, v)$. As mentioned before, if A and B are attempting to activate a node in L at the same time, the result will depend on the tie-breaking rule. If all edge probabilities are fixed, i.e., $\boldsymbol{\mu}$ is fixed, $r_S(\boldsymbol{\mu})$ is still submodular over S_A, so we can use a greedy algorithm as a $(1 - 1/e, 1)$-approximation oracle $\mathcal{O}_{\text{greedy}}$. Based on it, let us discuss the new offline optimization problem in Eq.(3.13) under our two tie-breaking rules: (1) $A > B$: since B will never influence nodes in R earlier than A in bipartite graphs, and A will always win the competition, from A's perspective, we can ignore S_B to choose S_A. In this case, all edge probabilities should take the maximum values: for all $i \in E$, μ_i equals to the upper bound of c_i, and we then use the oracle $\mathcal{O}_{\text{greedy}}$ to find S_A. (2) $B > A$: since A will never influence nodes in R earlier than B in bipartite graphs, and B will always win the competition, all out-edges of S_B, denoted as E_{S_B}, should take the minimum probabilities to maximize the influence spread of A. All the other edges in $E \backslash E_{S_B}$ should take the maximum probabilities. Formally, for all $i \in E_{S_B}$, μ_i equals to the lower bound of c_i; for all $i \in E \backslash E_{S_B}$, μ_i equals to the upper bound of c_i. We then use the oracle $\mathcal{O}_{\text{greedy}}$ to find S_A. To sum up, in bipartite graphs, $r_S(\boldsymbol{\mu})$ is optimized by pre-determining $\boldsymbol{\mu}$ based on the tie-breaking rule, and then using the greedy algorithm to get a $(1 - 1/e, 1)$-approximation solution. Since the time complexity of influence computation in the bipartite graph is $O(m)$, the time complexity of the offline algorithm is equal to that of the greedy algorithm, $O(kmn)$.

General Graph

GraphWe The competitive propagation in the general graph is much more complicated, so it is hard to pre-determine all edge probabilities as in the bipartite graph case. However, we have a key observation:

Lemma 3.4. *When fixing the seed set $S = \{S_A, S_B\}$, reward $r_S(\boldsymbol{\mu})$ has a linear relationship with each μ_i (when other μ_j's with $j \neq i$ are fixed). This implies that the optimal solution for the optimization problem in Eq.*(3.13) *must occur at the*

boundaries of the intervals c_i's.

Proof. We can expand $r_S(\boldsymbol{\mu})$ based on the live-edge graph model (Chen et al., 2013a):

$$r_S(\boldsymbol{\mu}) = \sum_L |\Gamma_A(L,S)| \cdot \Pr(L) = \sum_L |\Gamma_A(L,S)| \prod_{e\in E(L)} \mu_e \prod_{e\notin E(L)} (1-\mu_e), \qquad (3.47)$$

where L is one possible live-edge graph (each edge $e \in E$ is in L with probability μ_e and not in L with probability $1-\mu_e$, and this is independent from other edges), $\Gamma_A(L,S)$ is the set of nodes activated by A from seed sets $S = \{S_A, S_B\}$ under live-edge graph L and $E(L)$ is the set of edges that appear in live-edge graph L. Eq.(3.47) shows that $r_S(\boldsymbol{\mu})$ is linear with each μ_i, so the optimal μ_i must take either the minimum or the maximum value in its range c_i. □

Lemma 3.4 implies that for any edge e not reachable from B seeds, it is safe to always take its upper bound value since it can only helps the propagation of A. This further suggests that if we only have a small number (e.g. $\log m$) of edges reachable from B, then we can afford enumerating all the boundary value combinations of these edges. For each such boundary setting $\boldsymbol{\mu}$, we can use the IMM algorithm (Tang et al., 2014) to design a $(1-1/e-\epsilon, 1-n^{-l})$-approximation oracle $\mathcal{O}_{\text{IMM}}$ with time complexity $T_{\text{IMM}} = O((k+l)(m+n)\log n/\epsilon^2)$. We discuss such graphs that satisfy the above condition in directed trees. Specifically, we consider the in-arborescence, where all edges point towards the root. For any node u in the in-arborescence, there only exists one path from u to the root; if u is selected as the seed node of B, it could only propagate via this path. Hence, if the depth of the in-arborescence is in the order of $O(\log m)$, the number of edges reachable from S_B would be $O(|S_B| \cdot \log m)$. In this case, we can use the IMM algorithm for $O(m^{|S_B|})$ combinations to obtain an approximate solution with time complexity $O(m^{|S_B|} \cdot T_{\text{IMM}})$. Examples of such in-arborescences with depth $O(\log m)$ could be the complete or full binary trees.

For general graphs, designing efficient approximation algorithms for the offline problem in Eq. (3.13) remains a challenging open problem, due to the joint optimization over S and $\boldsymbol{\mu}$ and the complicated function form of $r_S(\boldsymbol{\mu})$. Nevertheless, heuristic algorithms are still possible. In the experiment section, we employee the following heuristic with the $B > A$ tie-breaking rule: for all outgoing edges from B seeds, we set their influence probabilities to their lower bound values, while for the rest, we set them to their upper bound values. This setting guarantees that the

first-level edges from the seeds are always set correctly, no matter how we select A seeds. They do not guarantee the correctness of second or higher level edge settings in the cascade, but the impact of those edges to influence spread decays significantly, so the above choice is reasonable as a heuristic.

3.6.5 Proof of Theorem 3.9

Proof. The OCIM-ETC algorithm is described in Alg. 3.7. We utilize the following well-known tail bound in our proof.

Lemma 3.5. *(Hoeffding's Inequality) Let $X_1, \ldots, X_n$ be independent and identically distributed random variables with common support $[0, 1]$ and mean μ. Let $Y = X_1 + \ldots, +X_n$. Then for all $\delta \geq 0$,*

$$\mathbb{P}\left\{|Y - n\mu| \geq \delta\right\} \leq 2e^{-2\delta^2/n}.$$

Let $\hat{\boldsymbol{\mu}} = (\hat{\mu}_1, \ldots, \hat{\mu}_m)$ be the empirical mean of $\boldsymbol{\mu}$. Recall that oracle $\mathcal{O}$ takes $S_B^{(t)}$ and $\hat{\boldsymbol{\mu}}$ as inputs and outputs a solution $S^{(t)}$. Let us define event $\mathcal{F} = \left\{r_{S^{(t)}}(\hat{\boldsymbol{\mu}}) < \alpha \cdot \mathrm{opt}^{(t)}(\hat{\boldsymbol{\mu}})\right\}$, which represents that oracle $\mathcal{O}$ fails to output an α-approximate solution, and we know $\mathbb{P}(\mathcal{F}) < 1 - \beta$.

With the same definitions in Appendix 3.6.3, we can decompose the regret as:

$$\begin{aligned}
Reg_{\alpha,\beta}(T; \boldsymbol{\mu}) &\leq \lceil nN/k \rceil \cdot \Delta_{\max}^{(T)} + \sum_{t=T-\lceil nN/k \rceil+1}^{T} \Big[\alpha\beta \cdot \mathrm{opt}^{(t)}(\boldsymbol{\mu}) - \mathbb{E}\big[r_{S^{(t)}}(\hat{\boldsymbol{\mu}})\big]\Big] \\
&\leq \lceil nN/k \rceil \cdot \Delta_{\max}^{(T)} + \sum_{t=T-\lceil nN/k \rceil+1}^{T} \Big[\alpha\beta \cdot \mathrm{opt}^{(t)}(\boldsymbol{\mu}) - \beta \cdot \mathbb{E}\big[r_{S^{(t)}}(\hat{\boldsymbol{\mu}}) \mid \neg\mathcal{F}\big]\Big] \\
&\leq \lceil nN/k \rceil \cdot \Delta_{\max}^{(T)} + \sum_{t=T-\lceil nN/k \rceil+1}^{T} \Big[\alpha \cdot \mathrm{opt}^{(t)}(\boldsymbol{\mu}) - \mathbb{E}\big[r_{S^{(t)}}(\hat{\boldsymbol{\mu}}) \mid \neg\mathcal{F}\big]\Big].
\end{aligned} \tag{3.48}$$

Next, let us rewrite the TPM condition in Theorem 3.5. For any S, $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$, we

have

$$
\begin{aligned}
|r_S(\boldsymbol{\mu}) - r_S(\boldsymbol{\mu}')| &\leq C \sum_{i\in[m]} p_i^S(\boldsymbol{\mu})|\mu_i - \mu_i'| \\
&\leq C \sum_{i\in[m]} |\mu_i - \mu_i'| \\
&\leq Cm \cdot \max_{i\in[m]} |\mu_i - \mu_i'|, \qquad (3.49)
\end{aligned}
$$

where C is the maximum number of nodes that any one node can reach in graph G. Let $S_{\boldsymbol{\mu}}^{*,t}$ denote the optimal action for $\boldsymbol{\mu}$ in round t. Under $\neg\mathcal{F}$, we have

$$
\begin{aligned}
r_{S^{(t)}}(\hat{\boldsymbol{\mu}}) &\geq \alpha \cdot r_{S_{\hat{\boldsymbol{\mu}}}^{*,t}}(\hat{\boldsymbol{\mu}}) \\
&\geq \alpha \cdot r_{S_{\boldsymbol{\mu}}^{*,t}}(\hat{\boldsymbol{\mu}}) \\
&\geq \alpha \cdot r_{S_{\boldsymbol{\mu}}^{*,t}}(\boldsymbol{\mu}) - \alpha \cdot Cm \cdot \max_{i\in[m]} |\mu_i - \hat{\mu}_i| \\
&\geq r_{S^{(t)}}(\boldsymbol{\mu}) + \Delta_{S^{(t)}}^{(t)} - \alpha \cdot Cm \cdot \max_{i\in[m]} |\mu_i - \hat{\mu}_i|, \qquad (3.50)
\end{aligned}
$$

where the third inequality is due to Eq.(3.49). Combining Eq.(3.49) and Eq.(3.50) together, we have

$$
\begin{aligned}
\Delta_{S^{(t)}}^{(t)} &\leq r_{S^{(t)}}(\hat{\boldsymbol{\mu}}) - r_{S^{(t)}}(\boldsymbol{\mu}) + \alpha \cdot Cm \cdot \max_{i\in[m]} |\mu_i - \hat{\mu}_i| \\
&\leq (1+\alpha) \cdot Cm \cdot \max_{i\in[m]} |\mu_i - \hat{\mu}_i|. \qquad (3.51)
\end{aligned}
$$

Let us define $\delta_0 := \frac{\Delta_{\min}^{(T)}}{2Cm}$. If $\max_{i\in[m]} |\mu_i - \hat{\mu}_i| < \delta_0$, then we know $S^{(t)}$ is at least an α-approximate solution, such that $\Delta_{S^{(t)}}^{(t)} = 0$. Then the regret in Eq.(3.48) can be written as

$$
\begin{aligned}
Reg_{\alpha,\beta}(T;\boldsymbol{\mu}) &\leq \lceil nN/k \rceil \cdot \Delta_{\max}^{(T)} + \Big(T - \lceil nN/k \rceil\Big) \cdot 2m \exp(-2N\delta_0^2) \cdot \Delta_{\max}^{(T)} \\
&\leq \Big(\lceil nN/k \rceil + T \cdot 2m \exp(-2N\delta_0^2)\Big) \cdot \Delta_{\max}^{(T)}. \qquad (3.52)
\end{aligned}
$$

The first inequality is obtained by applying the Hoeffding's Inequality (Lemma 3.5) and union bound to the event $\max_{i\in[m]} |\mu_i - \hat{\mu}_i| \geq \delta_0$. Now we need to choose an optimal N that minimizes Eq.(3.52). By taking $N = \max\left\{1, \frac{1}{2\delta_0^2} \ln \frac{4kmT\delta_0^2}{C}\right\} =$

$\max\left\{1, \frac{2C^2m^2}{(\Delta_{\min}^{(T)})^2}\ln(\frac{kT(\Delta_{\min}^{(T)})^2}{C^3m})\right\}$, when $\Delta_{\min}^{(T)} > 0$, we can get the distribution-dependent bound

$$\mathrm{Reg}_{\alpha,\beta}(T;\boldsymbol{\mu}) \leq \frac{2C^2m^2n\Delta_{\max}^{(T)}}{k(\Delta_{\min}^{(T)})^2}\left(\max\left\{\ln\left(\frac{kT(\Delta_{\min}^{(T)})^2}{C^2mn}\right), 0\right\}+1\right)+\frac{n}{k}\Delta_{\max}^{(T)}, \quad (3.53)$$

Next, let us prove the distribution-independent bound. Let $\mathcal{N}$ denote the event that $|\hat{\mu}_i - \mu_i| \leq \sqrt{\frac{2\ln T}{N}}$ for all $i \in [m]$. By the Hoeffding's Inequality and union bound, we have

$$\mathbb{P}\{\neg\mathcal{N}\} \leq m \cdot \frac{2}{T^4} \leq \frac{2}{T^3}. \quad (3.54)$$

When $\mathcal{N}$ holds, with Eq.(3.51), we have

$$\Delta_{S^{(t)}}^{(t)} \leq 2Cm \cdot \sqrt{\frac{2\ln T}{N}}, \quad (3.55)$$

and the regret in Eq.(3.48) can be written as

$$\begin{aligned} Reg_{\alpha,\beta}(T;\boldsymbol{\mu}) &\leq \lceil nN/k \rceil \cdot n + \sum_{t=T-\lceil nN/k\rceil+1}^{T} \Delta_{S^{(t)}}^{(t)} \\ &\leq \lceil nN/k \rceil \cdot n + O\left(T \cdot Cm \cdot \sqrt{\frac{\ln T}{N}}\right). \end{aligned} \quad (3.56)$$

We can choose N so as to (approximately) minimize the regret. For $N = (Cmk)^{\frac{2}{3}}n^{-\frac{4}{3}}T^{\frac{2}{3}}(\ln T)^{\frac{1}{3}}$, we obtain:

$$Reg_{\alpha,\beta}(T;\boldsymbol{\mu}) \leq O((Cmn)^{\frac{2}{3}}k^{-\frac{1}{3}}T^{\frac{2}{3}}(\ln T)^{\frac{1}{3}}). \quad (3.57)$$

To complete the proof, we need to consider both $\mathcal{N}$ and $\neg\mathcal{N}$. As shown in Eq.(3.54), the probability that $\neg\mathcal{N}$ occurs is very small, and we have:

$$\begin{aligned} Reg_{\alpha,\beta}(T;\boldsymbol{\mu}) &= \mathbb{E}\left[Reg_{\alpha,\beta}(T;\boldsymbol{\mu}) \mid \mathcal{N}\right] \cdot \mathbb{P}\{\mathcal{N}\} + \mathbb{E}\left[Reg_{\alpha,\beta}(T;\boldsymbol{\mu}) \mid \neg\mathcal{N}\right] \cdot \mathbb{P}\{\neg\mathcal{N}\} \\ &\leq \mathbb{E}\left[Reg_{\alpha,\beta}(T;\boldsymbol{\mu}) \mid \mathcal{N}\right] + T \cdot n \cdot O(T^{-3}) \\ &\leq O((Cmn)^{\frac{2}{3}}k^{-\frac{1}{3}}T^{\frac{2}{3}}(\ln T)^{\frac{1}{3}}). \end{aligned} \quad (3.58)$$

□

3.6.6 Proof of Theorem 3.10

Proof. As mentioned in Section 3.4.6, we need to introduce a virtual B seed node u_B, which connects to each existing node u with an unknown edge probability $p(u_B, u)$ equal to the probability of u being selected as a B seed. By adding these virtual nodes and edges, we get a new graph G' with $2n$ nodes and $m + n$ edges. Since S_B is fixed under G', we can follow the same steps in the proof of Theorem 3.5 to show the TPM condition holds under G'. Note that the maximum number of nodes that any one node can reach in G' is twice as that in the original graph G, so the new bounded smoothness coefficient $C = 2\tilde{C}$. □

Chapter 4

Competitive CMAB from the Multi-players' Perspective

4.1 Introduction

In this chapter, we study the competitive CMAB problem from the multi-players' perspective, where multiple players choose combinatorial actions on the same set of arms. Playing on the same arm incurs competition, which might lead to a potential loss of the reward. Our objective is to maximize the overall reward for all players. We first introduce the problem formulation of the centralized and distributed settings. In the centralized setting, we assume there exists a central controller making decisions for all players and also observing the feedback from all players. In the distributed setting, each player chooses her action individually only based on her own feedback. We then discuss the dynamic channel allocation application, where the competition comes from the collision of shared channels. We also discuss the application to the general online resource allocation problem, where the competition can be modeled as the constraints on resources.

4.2 Problem Formulation

We consider a learning game with M player. They play with an environment consisting of K random variables $X_1, \cdots, X_K$ called *base arms* following a joint distribution D over $[0,1]^K$. Distribution D is chosen by the environment from a

class of distributions $\mathcal{D}$ before the game starts. The players know $\mathcal{D}$ but not the actual distribution D in advance. The learning process runs in discrete rounds. We consider both centralized and distributed settings.

In the centralized setting, there is a central controller for all player. In round t, based on the feedback history from previous rounds, the central controller chooses a joint action $S^{(t)} = (S_1^{(t)}, \cdots, S_M^{(t)})$ from an action space $\mathcal{S}$ for all players, where $S_m^{(t)}$ is the action taken by player m. The environment draws an independent sample $X^{(t)} = (X_1^{(t)}, \cdots, X_K^{(t)})$ from the joint distribution D. When joint action $S^{(t)}$ is played on the environment outcome $X^{(t)}$, a random subset of arms $\tau_t \in [m]$ are triggered, and the outcomes of $X_i^{(t)}$ for all $i \in H(S^{(t)}, \tau_t)$ are observed as the feedback to the central controller, where H is the feedback function. τ_t may have additional randomness beyond the randomness of $X^{(t)}$. Let $D_{\text{trig}}(S, X)$ denote a distribution of the triggered subset of $[K]$ for given joint action S and an environment outcome X. We assume τ_t is drawn independently from $D_{\text{trig}}(S^{(t)}, X^{(t)})$. The central controller obtains a reward $R(S^{(t)}, X^{(t)}, \tau_t)$ fully determined by $S^{(t)}, X^{(t)}$ and τ_t. A learning algorithm aims at selecting actions $S^{(t)}$ over time based on past feedback to accumulate as much reward as possible. Note that the feedback function $H(S^{(t)}, \tau_t)$ and the reward function $R(S^{(t)}, X^{(t)}, \tau_t)$ are problem-specific and depend on the competition model.

In the distributed setting, players choose their own actions without any communication with each other. In round t, based on the feedback history of itself from previous rounds, each player m chooses an action $S_m^{(t)}$ from an action space $\mathcal{S}_m$. We can still consider the joint action of all players as $S^{(t)} = (S_1^{(t)}, \cdots, S_M^{(t)})$. The environment draws an independent sample $X^{(t)} = (X_1^{(t)}, \cdots, X_K^{(t)})$ from the joint distribution D. When joint action $S^{(t)}$ is played on the environment outcome $X^{(t)}$, a random subset of arms $\tau_t \in [m]$ are triggered, and the outcomes of $X_i^{(t)}$ for all $i \in H_m(S^{(t)}, \tau_t)$ are observed as the feedback to player m, where H_m is the feedback function of player m. τ_t may have additional randomness beyond the randomness of $X^{(t)}$. Let $D_{\text{trig}}(S, X)$ denote a distribution of the triggered subset of $[m]$ for given joint action S and an environment outcome X. We assume τ_t is drawn independently from $D_{\text{trig}}(S^{(t)}, X^{(t)})$. Each player m obtains a reward $R_m(S^{(t)}, X^{(t)}, \tau_t)$ fully determined by $S^{(t)}, X^{(t)}$ and τ_t. We denote the overall reward of all players as $R(S^{(t)}, X^{(t)}, \tau_t)$. Note that the relationship between the individual reward R_m and overall reward R is problem-specific, for example, R can be either the sum or the minmax of all R_m's. A learning algorithm deployed on all players aims at selecting

action $S^{(t)}$'s over time based on the past feedback to accumulate as much overall reward as possible.

For each base arm i, let $\mu_i = \mathbb{E}_{X\sim D}[X_i]$. Let $\boldsymbol{\mu} = (\mu_1, \cdots, \mu_m)$ denote the expectation vector of arms. We assume that the expected reward $\mathbb{E}[R(S, X, \tau)]$, where the expectation is taken over $X \sim D$ and $\tau \sim D_{\text{trig}}(S, X)$, is a function of the joint action S and the expectation vector $\boldsymbol{\mu}$ of the arms. Thus, we denote $r_{\boldsymbol{\mu}}(S) := \mathbb{E}[R(S, X, \tau)]$. The performance of a learning algorithm $\mathcal{A}$ is measured by its expected regret, which is the difference in expected cumulative reward between always playing the best action and playing actions selected by algorithm $\mathcal{A}$. Let $\text{opt}(\boldsymbol{\mu}) = \sup_{S^{(t)}\in\mathcal{S}} r_{\boldsymbol{\mu}}(S^{(t)})$ denote the expected reward of the optimal joint action in round t. We assume that there exists an offline oracle $\mathcal{O}$, which takes $\boldsymbol{\mu}$ as input and outputs a joint action $S^{\mathcal{O},(t)}$ such that $\Pr\{r_{\boldsymbol{\mu}}(S^{\mathcal{O},(t)}) \geq \alpha \cdot \text{opt}(\boldsymbol{\mu})\} \geq \beta$, where α is the approximation ratio and β is the success probability. Instead of comparing with the exact optimal reward, we take the $\alpha\beta$ fraction of it and use the following (α, β)-approximation *frequentist regret* for T rounds:

$$Reg^{\mathcal{A}}_{\alpha,\beta}(T; \boldsymbol{\mu}) = T \cdot \alpha \cdot \beta \cdot \text{opt}(\boldsymbol{\mu}) - \sum_{t=1}^{T} r_{\boldsymbol{\mu}}(S^{\mathcal{A},(t)}), \tag{4.1}$$

where $S^{\mathcal{A},(t)}$ is the joint action of all players chosen by algorithm $\mathcal{A}$ in round t.

4.3 Application in Dynamic Channel Allocation

4.3.1 Introduction

In the classical MAB problem, a player chooses one of a fixed set of arms and receives a reward based on this choice. The player aims to maximize her cumulative reward over multiple rounds, navigating a tradeoff between exploring unknown arms (to potentially discover an arm with higher rewards) and exploiting the best known arm (to avoid arms with low rewards). Most MAB algorithms use the history of rewards received from each arm to design optimized strategies for choosing which arm to play. They generally seek to prove that the *regret*, or the expected difference in the reward compared to the optimal strategy when all arms' reward distributions are known in advance, grows sub-linearly with the number of rounds.

Introducing Pre-observations

The classical MAB exploration-exploitation tradeoff arises because knowledge about an arm's reward can only be obtained by playing that arm. In practice, however, this tradeoff may be relaxed. [49], for example, suppose that at the end of each round, the player can pay a cost to observe the rewards of additional un-played arms, helping to find the best arm faster. In cascading bandits [5], players may choose multiple arms in a single round, e.g., if the "arms" are search results in a web search application.

In both examples above, the observations made in each round do not influence the choice of arms in that round. We introduce the **MAB problem with pre-observations**, where in each round, the player can pay to pre-observe the realized rewards of some arms before choosing an arm to play. For instance, one might play an arm with high realized reward as soon as it is pre-observed. Pre-observations can help to reconcile the exploration-exploitation tradeoff, but they also introduce an additional challenge: namely, **optimizing the order of the pre-observations**. This formulation is inspired by Cognitive Radio Networks (CRNs), where users can use wireless channels when they are unoccupied by primary users. In each round, a user can sense (pre-observe) some channels (arms) to check their availability (reward) before choosing a channel to transmit data (play). Sensing more arms leaves less time for data transmission, inducing a cost of making pre-observations.

In this pre-observation example, there are negative network effects when multiple players attempt to play the same arm: if they try to use the same wireless channel, for instance, the users "collide" and all transmissions fail. In multi-player bandit problems without pre-observations, players generally minimize these collisions by allocating themselves so that each plays a distinct arm with high expected reward. In our problem, the players must instead learn *ordered sequences* of arms that they should pre-observe, minimizing overlaps in the sequences that might induce players to play the same arm. Thus, one user's playing a sub-optimal arm may affect other users' pre-observations, leading to cascading errors. We then encounter a new challenge of **designing users' pre-observation sequences** to minimize collisions but still explore unknown arms. This problem is particularly difficult when **players cannot communicate or coordinate with each other** to jointly design their observation sequences. To the best of our knowledge, such *multi-player bandit problems with*

pre-observations have not been studied in the literature.

Applications

Although many MAB works take cognitive radios as their primary motivation [50, 51, 52], multi-player bandits with pre-observations could be applied to any scenario where users search for sufficiently scarce resources at multiple providers that are either acceptable (to all users) or not. We briefly list three more applications. First, users may sequentially bid in auctions (arms) offering equally useful items, e.g., Amazon EC2 spot instance auctions for different regions, stopping when they win an auction. Since these resources are scarce, each region may only be able to serve one user (modeling collisions between users). Second, in distributed caching, each user (player) may sequentially query whether one of several caches (arms) has the required file (is available), but each cache can only send data to one user at a time (modeling collisions). Third, taxis (players) can sequentially check locations (arms) for passengers (availability); collisions occur since each passenger can only take one taxi, and most locations (e.g., city blocks that are not next to transit hubs) would not have multiple passengers looking for a taxi at the same time.

Our Contributions

Our first contribution is to **develop an Observe-Before-Play (OBP) policy** to maximize the total reward of a single user via minimizing the cost spent on pre-observations. Our OBP policy achieves a regret bound that is logarithmic with time and quadratic in the number of available arms. It is consistent with prior results [53], and more easily generalizes to multi-player settings.

We next consider the multi-player setting. Unlike in the single-player setting, it is not always optimal to observe the arms with higher rewards first. We show that finding the offline optimal policy to maximize the overall reward of all players is NP-hard. However, we give conditions under which a greedy allocation that avoids user collisions is offline-optimal; in practice, this strategy performs well. Our second research contribution is then to **develop a centralized C-MP-OBP policy** that generalizes the OBP policy for a single user. Despite the magnified loss in reward when one user observes the wrong arm, we show that the C-MP-OBP policy can learn the arm rankings, and that its regret relative to the offline greedy

strategy is logarithmic with time and polynomial in the number of available arms and users. Our third research contribution is to **develop distributed versions of our C-MP-OBP policy, called D-MP-OBP and D-MP-Adapt-OBP**. Both algorithms assume no communication between players and instead use randomness to avoid collisions. Despite this lack of communication, both achieve logarithmic regret over time with respect to the collision-free offline greedy strategies defined in the centralized setting.

Our final contribution is to **numerically validate our OBP, C-MP-OBP, and D-MP-OBP policies on synthetic reward data and channel availability traces**. We show that all of these policies outperform both random heuristics and traditional MAB algorithms that do not allow pre-observations, and we verify that they have sublinear regret over time. We further characterize the effect on the achieved regret of varying the pre-observation cost and the distribution of the arm rewards.

Related Work

Multi-armed Bandit (MAB) problems have been studied since the 1950s [1, 54]. [55], for instance, propose a simple UCB1 policy that achieves logarithmic regret over time. Recently, MAB applications to Cognitive Radio Networks (CRNs) have attracted attention [56, 57], especially in multi-player settings [52, 58, 59, 60, 61] where users choose from the same arms (wireless channels). None of these works include pre-observations, though some [50, 51, 62] consider distributed settings. [53, 63] study the single-player MAB problem with pre-observations, but do not consider multi-player settings.

The proposed MAB with pre-observations in a single-player setting is a variant on cascading bandits [5, 17, 64]. The idea of pre-observations with costs is similar to the cost-aware cascading bandits proposed in [65] and contextual combinatorial cascading bandits introduced in [11]. However, in [65], the reward collected by the player can be negative if all selected arms have zero reward in one round; in our model, the player will get zero reward if all selected arms are unavailable. Moreover, most cascading bandit algorithms are applied to recommendation systems, where there is only a single player. To the best of our knowledge, we are the first to study MAB problems with pre-observations in multi-player settings.

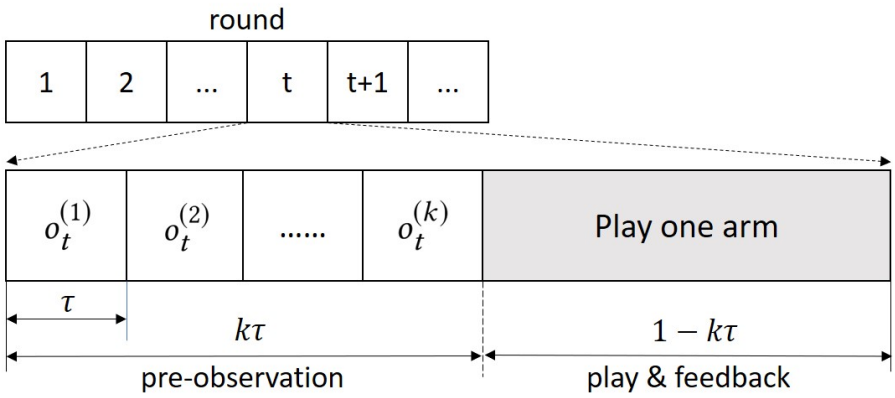


Figure 4.1: Illustration of pre-observations.

4.3.2 Single-player Setting

We consider a player who can pre-observe a subset of K arms and play one of them, with a goal of maximizing the total reward over T rounds. Motivated by the CRN scenario, we assume as in [59] an i.i.d. Bernoulli reward of each arm to capture the occupancy/vacancy of each channel (arm). Let $Y_{k,t} \overset{iid}{\sim} \text{Bern}(\mu_k) \in \{0,1\}$ denote the reward of arm k at round t, with expected value $\mu_k \in [0,1]$. As shown in Figure 4.1, in each round, the player chooses a pre-observation list $\boldsymbol{o}_t := (o_t^{(1)}, o_t^{(2)}, \ldots, o_t^{(K)})$, where $o_t^{(i)}$ represents the i^{th} arm to be observed at t and $\boldsymbol{o}_t$ is a permutation of $(1, 2, \ldots, K)$. The player observes from the first arm $o_t^{(1)}$ to the last arm $o_t^{(K)}$, stopping at and playing the first good arm (reward = 1) until the list exhausts. We denote the index of the last observed arm in $\boldsymbol{o}_t$ as $I(t)$, which is the first available arm in $\boldsymbol{o}_t$ or K if no arms are available. Pre-observing each arm induces a constant cost τ; in CRNs, this represents a constant time τ for sensing each channel's occupancy. We assume for simplicity that $0 < K\tau < 1$. The payoff received by the player at t then equals: $(1 - I(t)\,\tau)Y_{o_t^{(I(t))},t}$; if all the arms are bad (reward = 0) in round t, then the player will get zero reward for any $\boldsymbol{o}_t$. Given $\{\boldsymbol{o}_t\}_{t=1}^T$, we can then define the total realized and expected rewards received by the player in T rounds:

$$r(T) := \sum_{t=1}^{T} (1 - I(t)\,\tau) Y_{o_t^{(I(t))},t} \tag{4.2}$$

$$\mathbb{E}[r(T)] = \sum_{t=1}^{T} \sum_{k=1}^{K} \left\{ (1 - k\,\tau)\mu_{o_t^{(k)}} \prod_{i=1}^{k-1} (1 - \mu_{o_t^{(i)}}) \right\}, \tag{4.3}$$

where $\prod_{i=1}^{0}(1-\mu_{o_t^{(i)}}) := 1$. We next design an algorithm for choosing $\boldsymbol{o}_t$ at each round t to maximize $\mathbb{E}[r(T)]$. We assume $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$ without loss of generality and first establish the optimal offline policy:

Lemma 4.1. *The optimal offline policy $\boldsymbol{o}_t^*$ that maximizes the expected total reward is observing arms in the descending order of their expected rewards,* i.e., $\boldsymbol{o}_t^* = (1, 2, \ldots, K)$.

Algorithm 4.1 Observe-Before-Play UCB (OBP-UCB)

Initialization: Pull all arms once and update $n_i(t)$, $\overline{\mu}_i(t)$, $\hat{\mu}_i(t)$ for all $i \in [K]$
while t **do**
  $\boldsymbol{o}_t = \text{argsort}(\hat{\mu}_1(t), \hat{\mu}_2(t), \ldots, \hat{\mu}_K(t))$;
  for $i = 1 : K$ **do**
    Observe arm $o_t^{(i)}$'s reward $Y_{o_t^{(i)},t}$;
    $n_{o_t^{(i)}}(t+1) = n_{o_t^{(i)}}(t) + 1$;
    $\overline{\mu}_{o_t^{(i)}}(t+1) = (\overline{\mu}_{o_t^{(i)}}(t) n_{o_t^{(i)}}(t) + Y_{o_t^{(i)},t})/n_{o_t^{(i)}}(t+1)$;
    if $Y_{o_t^{(i)},t} = 1$ **then**
      Play arm i for this round;
      $n_{o_t^{(j)}}(t+1) = n_{o_t^{(j)}}(t)$ for all $j > i$;
      $\overline{\mu}_{o_t^{(j)}}(t+1) = \overline{\mu}_{o_t^{(j)}}(t)$ for all $j > i$;
      break;
    end if
  end for
  Update $\hat{\mu}_i(t)$ for all $i \in [K]$;
  $t = t + 1$;
end while

Given this result, we propose an UCB (upper confidence bound)-type online algorithm, Observe-Before-Play UCB (OBP-UCB), to maximize the cumulative expected reward without prior knowledge of the $\{\mu_k\}_{k=1}^K$. The OBP-UCB algorithm is formally described in Algorithm 4.1 and uses UCB values to estimate arm rewards as in traditional MAB algorithms [55]. Define $\overline{\mu}_i(t)$ as the sample average of μ_i up to round t and $n_i(t)$ as the number of times that arm i has been observed. Define $\hat{\mu}_i(t) := \overline{\mu}_i(t) + \sqrt{\frac{2\log t}{n_i(t)}}$ as the UCB value of arm i at round t. At each round, the player ranks all the arms i in descending order of $\hat{\mu}_i(t)$, and sets that order as $\boldsymbol{o}_t$. The player observes arms starting at $o_t^{(1)}$, stopping at the first good arm ($Y_{o_t^{(i)},t} = 1$) or when the list exhausts. She then updates the UCB values and enters the next

round. Since we store and update each arm's UCB value, the storage and computing overhead grow only linearly with the number of arms K.

We can define and bound the *regret* of this algorithm as the difference between the expected reward of the optimal policy (Lemma 4.1) and that of the real policy:

$$\begin{aligned} R(T) :=& \mathbb{E}[r^*(T)] - \mathbb{E}[r(T)] \\ =& \sum_{t=1}^{T} \sum_{k=1}^{K} \left\{ (1-k\,\tau)\mu_k \prod_{i=1}^{k-1}(1-\mu_i) - (1-k\,\tau)\mu_{o_t^{(k)}} \prod_{i=1}^{k-1}(1-\mu_{o_t^{(i)}}) \right\}. \end{aligned} \tag{4.4}$$

Theorem 4.1. *The total expected regret can be bounded as:* $\mathbb{E}[R(T)] \leq \sum_{i=1}^{K-1} \left\{ i\, W_i \sum_{j=i+1}^{K} [\frac{8\log T}{\Delta_{i,j}} + (1+\frac{\pi^2}{3})\Delta_{i,j}] \right\}$, *where* $W_k := (1-k\,\tau)\prod_{i=1}^{k-1}(1-\mu_i)$ *and* $\Delta_{i,j} := \mu_i - \mu_j$.

The expected regret $\mathbb{E}[R(T)]$ is upper-bounded in the order of $O(K^2 \log T)$, as also shown by [53]. However, our proof method is distinct from theirs and preserves the dependence on the arm rewards (through the W_i in Theorem 4.1). Since W_k converges to 0 as $k \to \infty$, we expect that the constant in our $O(K^2 \log T)$ bound will be small. Numerically, when there are more than 8 arms with expected rewards uniformly drawn from $(0, 1)$, our new regret bound is tighter than the result from [53] in 99% of our experiments. Moreover, unlike the analysis in [53], our regret analysis can be easily generalized to multi-player settings, as we show in the next section.

Algorithms with better regret order in T can be derived [63], but the regret bound of their proposed algorithm has a constant term (independent of T), $K^2\eta^2$, where $\eta = \prod_{i=1}^{K}(1-\mu_i)^{-1}$. This constant term is exponential in K so it can be significant if K is large. The same work also provides a lower bound in the order of $\Omega(K \log T)$ when the player can only choose less than K arms to pre-observe in each round.

4.3.3 Centralized Multi-player Setting

In the multi-player setting, we still consider K arms with i.i.d Bernoulli rewards; $Y_{k,t}$ denotes the realized reward of arm k at round t, with an expected value $\mu_k \in [0, 1]$. There are now $M \geq 1$ players ($M \leq K$) making decisions on which arms to observe and play in each round. We define a **collision** as two or more users playing the same arm in the same round, forcing them to share that arm's reward or even yielding

(a) Non-greedy optimal policy.

(b) Assigning arms.

Figure 4.2: Multi-player observation lists with expected rewards.

zero reward for all colliding players, e.g., in CRNs. In this setting, simply running the OBP-UCB algorithm on all players will lead to severe collisions, since all users may tend to choose the same observation list and play the same arm. To prevent this from happening, we first consider the case where a central controller can allocate different arms to different players.

At each round, the central controller decides pre-observation lists for all players; as in the single-player setting, each player sequentially observes the arms in its list and stops at the first good arm. The players report their observation results to the central controller, which uses them to choose future lists. A *policy* consists of a set of pre-observation lists for all players. Define $\boldsymbol{o}_{m,t} := (o_{m,t}^{(1)}, o_{m,t}^{(2)}, \ldots, o_{m,t}^{(i)}, \ldots)$ as the **pre-observation list** of player m at round t, where $o_{m,t}^{(i)}$ represents the i^{th} arm to be observed. The length of $\boldsymbol{o}_{m,t}$ can be less than K. Since collisions will always decrease the total reward, we only consider *collision-free policies*, i.e., those in which players' pre-observation lists are disjoint. Policies that allow collisions are impractical in CRNs as they waste limited transmission energy and defeat the purpose of pre-observations (sensing channel availability), which allow users to find an available channel without colliding with primary users. The expected overall reward of all players is then:

$$\mathbb{E}[r(T)] = \sum_{t=1}^{T} \sum_{m=1}^{M} \sum_{k=1}^{|\boldsymbol{o}_{m,t}|} \left\{ (1 - k\tau) \mu_{o_{m,t}^{(k)}} \prod_{i=1}^{k-1} (1 - \mu_{o_{m,t}^{(i)}}) \right\}. \tag{4.5}$$

Unlike in the single-player setting, the collision-free requirement now makes the expected reward for one player dependent on the decisions of other players. Intuitively, we would expect that a policy of always using better arms in earlier steps would perform well. We can in fact generalize Lemma 4.1 from the single-player setting:

Lemma 4.2. *Given a pre-observation list $\boldsymbol{o}_{m,t}$ for time t, player m maximizes its expected reward at time t by observing the arms in descending order of their rewards.*

With Lemma 4.2, we can consider the offline optimization of the centralized multi-player bandits problem. With the full information of expected rewards of all arms, i.e., $\{\mu_i\}_{i=1}^K$, the central controller allocates disjoint arm sets to different players, aiming to maximize the expected overall reward shown in (4.5). We show in Theorem 4.2 that the offline problem is NP-hard.

Theorem 4.2. *The offline problem of our centralized multi-player setting is NP-hard.*

Proof. Define $x_{ij} = 1$ if the central controller allocates arm j to player i and 0 otherwise. The offline optimization problem can be formulated as:

$$\begin{aligned} \max \quad & \sum_{i=1}^{M}\sum_{j=1}^{K}\Big\{\big[1-(\sum_{k<j}x_{ik}+1)\tau\big]x_{ij}\mu_j\prod_{k<j}(1-x_{ik}\mu_k)\Big\} \\ \text{s.t.} \quad & x_{ij}\in\{0,1\}, \\ & \sum_{i=1}^{M}x_{ij}\le 1,\ j=1,\dots,K, \end{aligned}$$

where we define $\sum_{\emptyset} := 0$ and $\prod_{\emptyset} := 1$. We show the Weapon Target Assignment (WTA) problem [66] with identical targets, which is NP-hard [67], can be reduced in polynomial time to a special case of our problem with $\tau = 0$: The WTA problem with identical targets aims to maximize the sum of expected damage done to all targets (mapped to be players), each of which can be targeted by possibly multiple weapons (mapped to be channels), where each weapon can only be assigned to at most one target and weapons of the same type have the same probability (mapped to be μ_k) to successfully destroy any target. Then, it is equivalent to maximizing the expected reward of all players when $\tau = 0$ in our problem. □

Although it is hard to find the exact offline optimal policy, Lemma 4.2 suggests that a **collision-free greedy** policy, which we also refer to as a *greedy policy*, might

be closed to the optimal one. We first define the i^{th} **observation step** in a policy as the set of arms in the i^{th} positions of the players' observation lists, denoted by $\boldsymbol{s}_{i,t} := (o_{1,t}^{(i)}, o_{2,t}^{(i)}, \ldots, o_{M,t}^{(i)})$ for each round t. We define a *greedy policy* as one in which at each observation step, the players greedily choose the arms with highest expected rewards from all arms not previously observed. Formally, assuming without loss of generality that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$, in the ith observation step, players should observe different arms from the set $\boldsymbol{s}_{i,t} = \{(i-1)M+1, (i-1)M+2, \ldots, iM\}$. In the simple **greedy-sorted policy**, for instance, player m will choose arm $(i-1)M+m$ in the i^{th} observation step. A potentially better candidate is the **greedy-reverse policy**: at each observation step, arms are allocated to players in the reverse order of the probability they observe an available arm from previous observation steps. Formally, in the ith observation step, arm $(i-1)M+j$ is assigned to the player m with the jth highest value of $\Pi_{l=1}^{i-1}(1-\mu_{\boldsymbol{o}_{m,t}^{(l)}})$, or the probability player m has yet not found an available arm. Experiments show that when there are 3 players and 9 arms with expected rewards uniformly drawn from $(0,1)$, the greedy-reverse policy is the optimal greedy policy 90% of the time. In fact,

Lemma 4.3. *When $K \leq 2M$, the optimal policy is the greedy-reverse policy.*

In general, the optimal policy may not be the greedy-reverse one, or even a greedy policy. Figure 4.2a shows such a counter-intuitive example. In this example, player 1 should choose the arm with 0.15 expected reward, not the one with 0.25 expected reward, in step 2. Player 1 should reserve the higher-reward arm for player 3 in a later step, as player 3 has a lower chance of finding a good arm in steps 1 or 2. In practice, we expect these examples to be rare; they occur less than 30% of the time in simulation. Thus, we design an algorithm that allocates arms to players according to a specified greedy policy (e.g., greedy-sorted) and bound its regret.

We propose an UCB-type online algorithm, **Centralized Multi-Player Observe-Before-Play** (C-MP-OBP), to learn a greedy policy without prior knowledge of the expected rewards $\{\mu_k\}_{k=1}^K$. The C-MP-OBP algorithm is described in Algorithm 1, generalizing the single-player setting. To simplify the discussion, we assume $K/M = L$, i.e., each player will have an observation list of the same length, L, when using a greedy policy. Note that if K is not a multiple of M, we can introduce virtual arms with zero rewards to ensure $K/M = L$. At each round t, the central controller ranks all the arms in the descending order of $\hat{\mu}_i(t)$, the UCB value of arm i

Algorithm 4.2 Centralized Multi-Player OBP (C-MP-OBP)

Initialization: Pull all arms once and update $n_i(t)$, $\overline{\mu}_i(t)$, $\hat{\mu}_i(t)$ for all $i \in [K]$
while t **do**
  $\boldsymbol{\alpha} = \text{argsort}(\hat{\mu}_1(t), \hat{\mu}_2(t), \ldots, \hat{\mu}_K(t))$;
  for $i = 1 : L$ **do**
    $\boldsymbol{s}_{i,t} = \boldsymbol{\alpha}[(i-1) * M + 1 \ : \ i * M]$
  end for
  for $m = 1 : M$ **do**
    for $i = 1 : L$ **do**
      Observe arm $\boldsymbol{s}_{i,t}[m]$'s reward $Y_{\boldsymbol{s}_{i,t}[m],t}$;
      $n_{\boldsymbol{s}_{i,t}[m]}(t+1) = n_{\boldsymbol{s}_{i,t}[m]}(t) + 1$;
      $\overline{\mu}_{\boldsymbol{s}_{i,t}[m]}(t+1)$
      $= \left(\overline{\mu}_{\boldsymbol{s}_{i,t}[m]}(t) + Y_{\boldsymbol{s}_{i,t}[m],t}\right) / n_{\boldsymbol{s}_{i,t}[m]}(t+1)$;
      if $Y_{\boldsymbol{s}_{i,t}[m],t} = 1$ **then**
        Player m plays arm $\boldsymbol{s}_{i,t}[m]$ for this round;
        $n_{\boldsymbol{s}_{j,t}[m]}(t+1) = n_{\boldsymbol{s}_{j,t}[m]}(t)$ for all $j > i$;
        $\overline{\mu}_{\boldsymbol{s}_{j,t}[m]}(t+1) = \overline{\mu}_{\boldsymbol{s}_{j,t}[m]}(t)$ for all $j > i$;
        break;
      end if
    end for
  end for
  Update $\hat{\mu}_i(t)$ for all $i \in [K]$;
  $t = t + 1$;
end while

at round t, and saves that order as $\boldsymbol{\alpha}$. Then it sets the first M arms in $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}[1:M]$, as $\boldsymbol{s}_{1,t}$, the second M arms in $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}[M+1:2M]$ as $\boldsymbol{s}_{2,t}$, and so on, assigning the arms in each list to players according to the specified greedy policy. Each player m's observation list is then $\boldsymbol{o}_{m,t} = (\boldsymbol{s}_{1,t}[m], \ldots, \boldsymbol{s}_{L,t}[m])$. At the end of this round, the central controller aggregates all players' observations to update the UCB values and enter the next round.

We define the *regret*, $R(T) := \mathbb{E}[r^*(T)] - \mathbb{E}[r(T)]$, as the difference between the expected reward of the target policy and that of C-MP-OBP algorithm:

$$R(T) = \sum_{t,m,k=1}^{T,M,L} \left\{ (1-k\tau)\mu_{(k-1)M+m} \prod_{i=1}^{k-1}(1-\mu_{(i-1)M+m}) - (1-k\tau)\mu_{o_{m,t}^{(k)}} \prod_{i=1}^{k-1}(1-\mu_{o_{m,t}^{(i)}}) \right\}. \tag{4.6}$$

Defining $c_\mu := \frac{\mu_{\max}}{\Delta_{\min}}$, we show the following regret bound:

Theorem 4.3. *The expected regret of C-MP-OBP is bounded by*

$$\mathbb{E}[R(T)] \leq c_\mu K^2(L^2+L)\left(\frac{8\log T}{\Delta_{\min}} + (1+\frac{\pi^2}{3})\Delta_{\max}\right),$$

where $\Delta_{\max} = \max_{i<j} \mu_i - \mu_j$, $\Delta_{\min} = \min_{i<j} \mu_i - \mu_j$.

The expected regret $E[R(T)]$ is upper bounded in the order of $O(K^2L^2\log T)$, compared to $O(K^2\log T)$ in the single-player setting. Thus, we incur a "penalty" of L^2 in the regret order, due to sub-optimal pre-observations' impact on the subsequent pre-observations of other users. We note that, if pre-observations are not allowed, we can adapt the proof of Theorem 4.3 to match the lower bound of $O(KM\log T)$ given by [51].

4.3.4 Distributed Multi-player Setting

We finally consider the scenario without a central controller or any means of communication between players. In the CRN setting, for instance, small Internet-of-Things devices may not be able to tolerate the overhead of communication with a central server. The centralized C-MP-OBP policy is then infeasible, and specifying a collision-free policy is difficult, as the players make their decisions independently. We propose a **Distributed Multi-Player Observe-Before-Play** (D-MP-OBP) online algorithm in which each player distributedly learns a "good" policy that effectively avoids collisions with others. Specifically, it converges to one of the offline collision-free greedy policies that we defined in Section 4.3.3; we then show that D-MP-OBP can be adapted to achieve a pre-specified greedy policy, e.g., greedy-reverse. To facilitate the discussion, we define $\eta_k^{(t)}$ as an indicator that equals 1 if more than one player plays arm k in round t and 0 otherwise. As in the centralized setting, $o_{m,t}^{(k)}$ denotes the k^{th} arm in player m's observation list at round t.

Algorithm 4.3 Distributed Multi-Player OBP (D-MP-OBP)

Initialization: Pull all arms once and update $n_i(t)$, $\overline{\mu}_i(t)$, $\hat{\mu}_i(t)$ for all $i \in [K]$
while t **do**
  $\boldsymbol{\alpha} = \text{argsort}(\hat{\mu}_1(t), \hat{\mu}_2(t), \ldots, \hat{\mu}_K(t))$;
  for $i = 1 : L$ **do**
    $\boldsymbol{s}_{i,t} = \boldsymbol{\alpha}[(i-1) * M + 1 \ : \ i * M]$
  end for
  for $i = 1 : L$ **do**
    if $m_i^* = 0$ OR $m_i^* \notin \boldsymbol{s}_{i,t}$ **then**
      The player uniformly at random selects an arm from $\boldsymbol{s}_{i,t}$ to observe and record the index of the chosen arm as m_i^*;
    end if
    Observe the reward $Y_{\boldsymbol{s}_{i,t}[m_i^*],t}$;
    $n_{\boldsymbol{s}_{i,t}[m_i^*]}(t+1) = n_{\boldsymbol{s}_{i,t}[m_i^*]}(t) + 1$;
    $\overline{\mu}_{\boldsymbol{s}_{i,t}[m_i^*]}(t+1)$
    $= \left(\overline{\mu}_{\boldsymbol{s}_{i,t}[m_i^*]}(t) + Y_{\boldsymbol{s}_{i,t}[m_i^*],t}\right) / n_{\boldsymbol{s}_{i,t}[m_i^*]}(t+1)$;
    if $Y_{\boldsymbol{s}_{i,t}[m_i^*],t} = 1$ **then**
      The player plays arm $\boldsymbol{s}_{i,t}[m*]$ for this round;
      $n_{\boldsymbol{s}_{j,t}[m*]}(t+1) = n_{\boldsymbol{s}_{j,t}[m*]}(t)$ for all $j > i$;
      $\overline{\mu}_{\boldsymbol{s}_{j,t}[m*]}(t+1) = \overline{\mu}_{\boldsymbol{s}_{j,t}[m*]}(t)$ for all $j > i$;
      break;
    end if
  end for
  if a collision occurs **then**
    Update $m_i^* = 0$;
  end if
  Update $\hat{\mu}_i(t)$ for all $i \in [K]$;
  $t = t + 1$;
end while

The D-MP-OBP algorithm is shown in Algorithm 4.3. As in the C-MP-OBP algorithm, in each round, each player independently updates its estimate of the expected reward (μ_k) for each arm k using the UCB of μ_k. Each player then sorts the estimated $\{\mu_k\}_{k=1}^K$ into descending order and groups the K arms into L sets. We still use $\boldsymbol{s}_{i,t}$ to denote the list of arms that the players observe in step i at round t. Since users may have different lists $\boldsymbol{s}_{i,t}$ depending on their prior observations, we cannot simply allocate the arms in $\boldsymbol{s}_{i,t}$ to users. Instead, the users follow a randomized strategy in each step i at round t. If there was a collision with another player on

arm i at round $t-1$ or the arm chosen in round $t-1$ does not belong to her own set $\boldsymbol{s}_{i,t}$, then the player uniformly at random chooses an arm from her $\boldsymbol{s}_{i,t}$ to observe. Otherwise, the player observes the same arm as she did in step i in round $t-1$. If the arm is observed to be available, the player plays it and updates the immediate reward and the UCB of the arm. Otherwise, she continues to the next observation step. Note that this policy does not require any player communication.

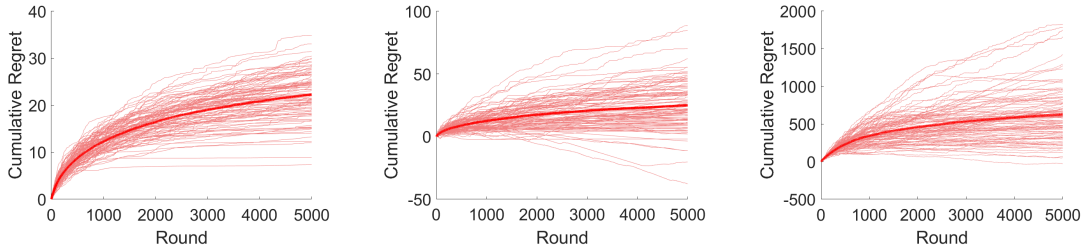
To evaluate D-MP-OBP, we define a performance metric, Loss(T), to be the maximum difference in total reward over T rounds between any collision-free greedy policy and the reward achieved by D-MP-OBP. Thus, unlike the regret $\mathbb{E}[R(T)]$ defined for our C-MP-OBP policy, $\mathbb{E}[\text{Loss}(T)]$ does not target a specific greedy policy. Moreover, unlike C-MP-OBP, our D-MP-OBP algorithm provides fairness in expectation for all players, as they have equal opportunities to use the best arms in each observation step.

Theorem 4.4. *The total expected loss, $\mathbb{E}[Loss(T)]$, of our distributed algorithm D-MP-OBP is logarithmic in T.*

We finally define the **D-MP-Adapt-OBP** algorithm, which adapts Algorithm 4.3 to steer the players towards a specific policy by adding a small extra term for each player. We define a function $f(\cdot)$ for each player to map the arm chosen in the first observation step to the arm chosen in the following steps given the predictions of each μ_k. With some abuse of notation, we define $o^l_{m,t}$ as the arm chosen by player m for step l in round t. The function f then steers the players to the collision-free greedy policy given by $o^{l+1}_{m,t} = f(o^l_{m,t}, \{\hat{\mu_k}(t)\}_{k=1}^K), \forall l = 1, ..., L-1$ for each player m; we define the regret with respect to this policy.

We can view the function f as replacing the player index in the centralized setting with the relative ranking of the arm chosen by this player in prior observation steps. As an example, the greedy-sorted policy used in Section 4.3.3 is equivalent to: (1) letting players choose different arms, and (2) the player that chooses the arm in position m continuing to choose the arm with the m^{th} best reward of its set $\boldsymbol{s}_{i,t}$ in each subsequent step. Thus, we can steer the players to specific observation lists within a given collision-free greedy policy. Their decisions then converge to the specified policy.

Theorem 4.5. *The expected regret, $\mathbb{E}[R(T)]$ of our distributed algorithm D-MP-Adapt-OBP is logarithmic in T.*



(a) OBP-UCB. **(b)** C-MP-OBP. **(c)** D-MP-OBP.

Figure 4.3: Sublinear regret in each setting.

Table 4.1: Average % reward improvements of OBP-UCB.

τ	single-opt	random	single-real	random-real
0.01	102%	5%	76%	6%
0.05	92%	34%	71%	47%
0.1	78%	140%	63%	245%

We observe from the proof of Theorem 4.5 that the regret is combinatorial in M but logarithmic in T, unlike the centralized multi-player setting's $O(K^2L^2 \log T)$ regret in Theorem 4.3. This scaling with M comes from the lack of coordination between players and the resulting collisions.

4.3.5 Experiments

We validate the theoretical results from Sections 4.3.2–4.3.4 with numerical simulations. We summarize our results as follows:

Sublinear regret: We show in Figure 4.3 that our algorithms in the single-player, multi-player centralized, and multi-player distributed settings all achieve a sublinear regret, respectively defined relative to the single-player offline optimal (Lemma 4.1), the greedy-sorted policy, and a collision-free-greedy-random policy that in each step greedily chooses the set of arms but randomly picks one collision-free allocation. Figure 4.3b shows our C-MP-OBP algorithm's regret is even negative for a few runs: by deviating from the greedy-sorted policy towards the true optimum, the C-MP-OBP algorithm may obtain a higher reward. The regret of D-MP-OBP in Figure 4.3c is larger than that of C-MP-OBP, likely due to collisions in the distributed setting.

Table 4.2: Average C-MP-OBP, D-MP-OBP % improvement.

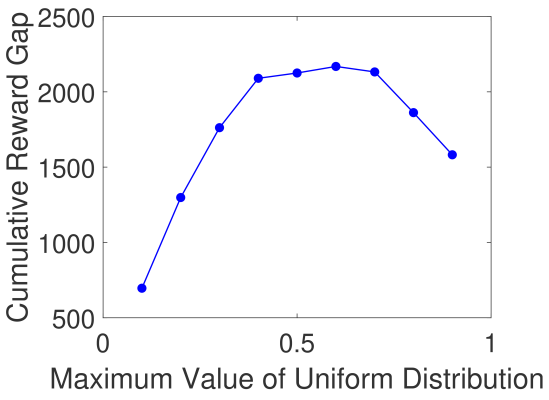
τ	single-opt	random	single-real	random-real
0.1	41%, 27%	7%, 39%	35%, 198%	4%, 30%
0.2	33%, 20%	15%, 47%	28%, 183%	10%, 36%
0.3	22%, 11%	30%, 60%	19%, 165%	20%, 47%

Superiority to baseline strategies: We show in Tables 4.1 and 4.2 that our algorithms consistently outperform two baselines, in both synthetic reward data ($K = 9$ arms with expected rewards uniformly drawn from $[0, 0.5]$ and $M = 3$ players for multi-player settings) and real channel availability traces [68]. Our first baseline is a **random heuristic** (called **random** for synthetic data and **random-real** for real data trace) in which users pre-observe arms uniformly at random and play the first available arm. Comparisons to this baseline demonstrate the value of strategically choosing the order of the pre-observations. Our second baseline is an **optimal offline single-observation policy** (**single-opt**), which allocates the arms with the M highest rewards to each player (in the single-player setting, $M = 1$). These optimal offline policies are superior to any learning-based policy with a single observation, so comparisons with this baseline demonstrate the value of pre-observations. When the rewards are drawn from a real data trace, they may no longer be i.i.d. Bernoulli distributed, so these offline policies are no longer truly "optimal." Instead, we take a **single-observation UCB algorithm** (**single-real**) as the baseline; this algorithm allocates the arms with the top M (≥ 1) highest UCB values to different users, and each player still observes and plays one such arm in each round.

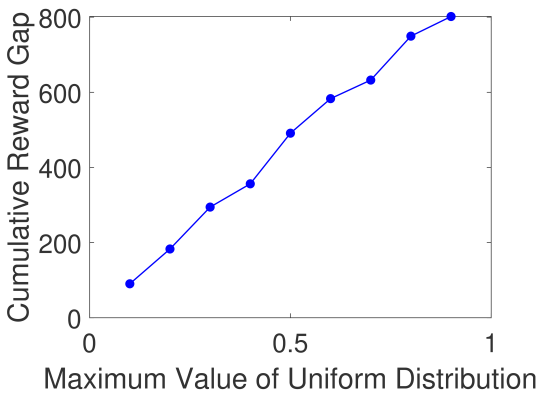
Tables 4.1 and 4.2 show the average improvements in the cumulative reward achieved by our algorithms over the baselines after 5000 rounds over 100 experiment repetitions with different τ. In each setting, increasing τ causes the improvement over the random baseline to increase: when τ is small, there is little cost to mis-ordered observations, so the random algorithm performs relatively well. Conversely, increasing τ narrows the reward gap with the single-observation baseline: as pre-observations become more expensive, allowing users to make them does not increase the reward as much.

Effect of μ: We would intuitively expect that increasing the average rewards μ_i would increase the reward gap with the random baseline: it is then more important to pre-observe "good" arms first, to avoid the extra costs from pre-observing occupied

arms. We confirm this intuition in each of our three settings. However, increasing the μ's does not always increase the reward gap with the single-observation baseline, since if the μ's are very low or very high, pre-observations are less valuable. When the μ's are small, the player would need to pre-observe several arms to find an available one, decreasing the final reward due to the cost of these pre-observations. When the μ's are large, simply choosing the best arm is likely to yield a high reward, and the pre-observations would add little value. Figures 4.4a and 4.4b plot the reward gap with respect to x (μ's are drawn from $U(0, x)$) : an increase in x increases the reward gap with the random baseline, but has a non-monotonic effect compared to the single-observation baseline. Similar trends are also found in multi-player settings. In Figures 4.5a–4.5d, we plot the reward gap with respect to μ in the centralized and distributed multi-player settings. As in the single-player setting, an increase in μ increases the reward gap with the random baseline, but has a non-monotonic effect compared to the single-observation baseline.



(a) Single-observation baseline.



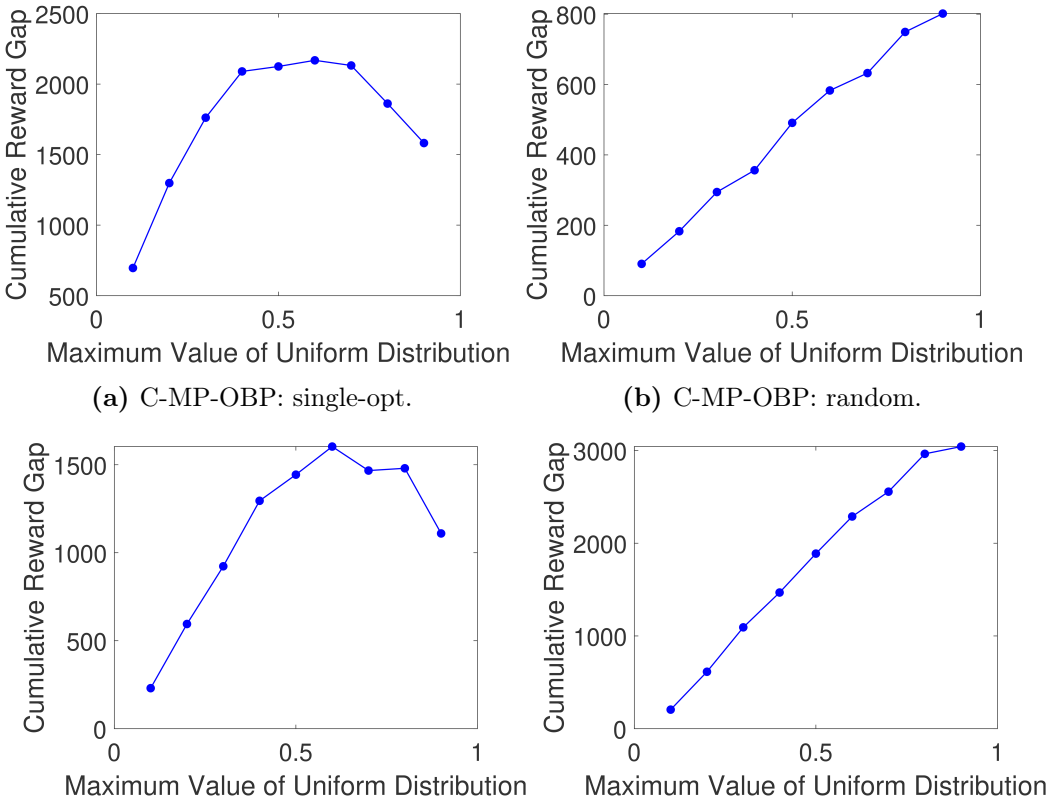
(b) Random baseline.

Figure 4.4: Average cumulative reward gaps in the single-player setting.

4.4 Application in Resource Allocation

4.4.1 Introduction

Resource allocation, which generally refers to the problem of distributing a limited budget among multiple entities, is a fundamental challenge that arises in many types of systems, including wireless networks, computer systems, and power grids.



(a) C-MP-OBP: single-opt. **(b)** C-MP-OBP: random.

(c) D-MP-OBP: single-opt. **(d)** D-MP-OBP: random.

Figure 4.5: Average cumulative reward gaps in the single-player, centralized multi-player, and distributed multi-player settings.

Generally, the entity in charge of distributing the budget wishes to do so in an "optimal" manner, where "optimality" may be defined according to a variety of objectives. In this work, we consider an *online* version of the resource allocation problem, where the objective is not known a priori but can be learned over time based on feedback from the users to whom the budget is allocated. In the online resource allocation problem, a decision maker (agent) must repeatedly distribute its available budgets among different resources (users). Each resource will generate a random reward based a general reward function of the allocated budget and other, unknown factors. As the resource allocation task is repeated over time, the decision maker can gather information about the reward functions and the unknown distributions from observed reward feedback. Its goal is to maximize the cumulative total reward, or equivalently, minimize the cumulative *regret* compared to the total achievable award

if the reward distributions were known.

In this section, we introduce offline and online versions of the general resource allocation problem without specifying the exact forms of the reward functions. We assume that the obtained rewards of different resources are independent from each other and that the agent has to balance the tradeoff between exploration and exploitation: as the total budget is limited, the agent needs to intelligently allocate it to not only the resources that may provide high rewards, but also those that have not been tried many times yet. We consider both discrete and continuous budget allocations, which can be applied to different real-world applications. For example, discrete budget allocation can be used in the case of allocating computation tasks with the same job size to different servers, while continuous budget allocation can model the power allocation problem on wireless channels. In these examples, uncertainty in the reward functions can result from unknown competition for server resources and unknown channel conditions for individual users, respectively.

We propose two algorithms, CUCB-DRA and CUCB-CRA, for online discrete and continuous resource allocation, respectively. We adopt the Combinatorial Multi-arm Bandit (CMAB) framework [2] for the online Discrete Resource Allocation (DRA) problem. The proposed CUCB-DRA algorithm considers the action "allocating a budget to resource k" as a base arm (k, a). By introducing these base arms, CUCB-DRA does not have to learn the exact form of the reward functions of different resources, and only needs to maintain the upper confidence bounds (UCBs) on the expected rewards of playing the (k, a)'s, which are updated by the obtained rewards from each resource in each round. We prove that CUCB-DRA achieves logarithmic regret with the number of rounds T. For the online Continuous Resource Allocation (CRA) problem, as the action space becomes infinite, we cannot directly apply the CMAB framework. We propose a CUCB-CRA algorithm integrating CMAB with fixed discretization [69] that splits the continuous action space into a discrete one. This discretization technique relies on a Lipschitz condition, which is satisfied by many types of real reward functions. We decompose the cumulative regret into the learning regret and the discretization error, then choose the optimized discretization granularity to minimize the sum of them. We show that CUCB-CRA achieves logarithmic regrets with T. In addition, since the rewards obtained from the same resource under different budget allocations are often correlated in practical applications, we propose a correlated combinatorial bandit algorithm [13]

that can achieve improved regret bounds than correlation-agnostic algorithms. We demonstrate the effectiveness of all proposed algorithms through experiments in wireless applications.

Related Work. The classical resource allocation problem has been extensively studied for decades [70, 71, 72, 73]. Recently, the online version of the resource allocation problem has attracted much attention [74, 75, 76, 77]. For example, [75] introduced the online linear resource allocation problem where the reward functions are assumed to be linear, while [76] studied online resource allocation with censored feedback. [77] considered the online resource allocation problem with concave reward functions. All of these previous works assumed specific types of reward functions, while in this thesis, we introduce an online resource allocation framework with *general* reward functions and show that combinatorial bandit techniques can be used to achieve logarithmic solution regret. To best of our knowledge, we are also the first to formally model the presence of discrete and continuous budgets in online resource allocation problems. Our proposed CUCB-DRA is based on the CUCB algorithm in [2]. However, CUCB was designed for a binary action space, which we extend to the finite discrete space, by introducing a new definition of base arms for the online discrete resource allocation problem. CUCB-CRA further extends the action space to an infinite continuous space, by combining the idea of CUCB with that of fixed discretization in [69]. Its regret analysis relies on the Lipschitz condition defined in [69] and the 1-norm bounded smoothness condition defined in [3].

The idea of capturing correlations in reward across different arms was previously studied in the context of classical multi-armed bandits, i.e., the setting where only one base arm is played in each round, in [78, 79]. Another closely related line of work, where only one base arm is played in each round, is that of structured bandits [80, 81, 82], where the mean rewards corresponding to different base arms are related to one another through a hidden parameter θ. While the mean rewards between different arms are related to one another in structured bandits, they are not necessarily correlated. Due to this, the correlated bandit framework [78, 79] fits better to the problem setting of online resource allocation where reward realizations are known to be correlated. We extend this idea of correlated bandits to the combinatorial bandit framework and propose the correlated UCB algorithm for online resource allocation. The extension is non-trivial as the classical multi-armed bandit and combinatorial bandit often require different design of algorithms and

regret analysis due to the selection of multiple base arms within provided constraints as opposed to the selection of the single base arm in each round. Upon doing so, we are able to exploit the correlations to obtain significant performance improvements as demonstrated in later sections. To the best of our knowledge, this is the first work to show that $O(1)$ regret can be achieved in certain online resource allocation problems.

4.4.2 Problem Formulation

In this section, we formulate the problem of allocating a fixed budget to multiple resources. We first introduce the offline problem and then consider the online setting.

Offline Setting

We consider a general resource allocation problem where a decision maker has access to K different types of resources. The decision maker has to split a total amount of Q divisible budget and allocate a_k amount of the budget to each resource $k \in [K]$. For example, the budget may represent compute time, and each resource may represent a user of a particular server. We consider a general reward function $f_k(a_k, X_k)$ of each resource k, where a_k is the allocated budget and X_k is a random variable that reflects the random fluctuation of the generated reward. Notice that the allocated budget a_k's could be either discrete (e.g., $a_k \in \mathbb{N}$) or continuous (e.g., $a_k \in \mathbb{R}_{\geq 0}$), and we denote the feasible action space of a_k as $\mathcal{A}$. For the offline setting, we assume the distributions of X_k for all $k \in [K]$ are known in advance and denote them as D_k. Our goal is to maximize the expected total reward collected from all resources. This can be formulated as the optimization problem below:

$$\begin{aligned} \underset{a_k}{\text{maximize}} \quad & \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_k, X_k)\right] \\ \text{subject to} \quad & \sum_{k=1}^{K} a_k \leq Q, a_k \in \mathcal{A} \end{aligned} \tag{4.7}$$

With different reward functions $f_k(a_k, X_k)$ and action spaces $\mathcal{A}$, the hardness of this offline optimization problem varies. For example, it becomes a convex optimization problem if $f_k(a_k, X_k)$ is convex over a_k and $\mathcal{A}$ is a convex set for all $k \in [K]$; at the

other extreme, it can also be a NP-hard combinatorial optimization problem when $\mathcal{A}$ is a discrete set. We will not specify the exact form of the optimization problem, but only assume that there exists an offline approximation oracle that can give us an approximate solution with constant approximation ratio. More details of the offline oracle will be discussed in the next section.

Online Setting

Now we introduce the online version of the general resource allocation problem, which is a sequential decision making problem. In each round t, we allocate $a_{k,t}$ budget to resource k for all $k \in [K]$, subject to the total budget constraint, $\sum_{k=1}^{K} a_{k,t} \leq Q$. We then observe the semi-bandit feedback, which is the reward $f_k(a_{k,t}, X_{k,t})$ from each resource k, where $X_{k,t}$ is sampled from an unknown distribution D_k. The total obtained reward is $\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})$. Our goal is to accumulate as much total reward as possible through this repeated budget allocation over multiple rounds.

We denote the budget allocation to all resources at round t as $\boldsymbol{a}_t = (a_{1,t}, \cdots, a_{K,t})$ and the joint distribution of all independent $X_{k,t}$'s as $\boldsymbol{D} = (D_1, \cdots, D_K)$. The expected reward in round t can be defined as $r(\boldsymbol{a}_t, \boldsymbol{D}) = \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})\right]$. We consider a learning algorithm π that makes the budget allocation $\boldsymbol{a}_t^{\pi}$ for round t. We can then measure the performance of π by its (expected) regret, which is the difference in expected cumulative reward between always taking the best offline allocation and taking the budget allocation selected by algorithm π. Let $\text{opt}(\boldsymbol{D}) = \sup_{\boldsymbol{a}_t} r(\boldsymbol{a}_t, \boldsymbol{D})$ denote the expected total reward of the optimal allocation in round t. As discussed in the previous section, we assume that there exists an offline (α, β)-approximation oracle $\mathcal{O}$, which outputs an allocation $\boldsymbol{a}_t^{\mathcal{O}}$ such that $\Pr\{r(\boldsymbol{a}_t^{\mathcal{O}}, \boldsymbol{D}) \geq \alpha \cdot \text{opt}(\boldsymbol{D}))\} \geq \beta$, where α is the approximation ratio and β is the success probability. Instead of comparing with the exact optimal reward, we take the $\alpha\beta$ fraction of it and use the following (α, β)-approximation regret for T rounds:

$$Reg_{\alpha,\beta}^{\pi}(T; \boldsymbol{D}) = T \cdot \alpha \cdot \beta \cdot \text{opt}(\boldsymbol{D}) - \sum_{t=1}^{T} r(\boldsymbol{a}_t^{\pi}, \boldsymbol{D}), \tag{4.8}$$

In the next two sections, we give solution algorithms for our resource allocation problem that achieve sublinear (α, β)-approximation regrets when the action space $\mathcal{A}$ is discrete (Section 4.4.3) and continuous (Section 4.4.4).

4.4.3 Online Discrete Resource Allocation

In this section, we consider the online Discrete Resource Allocation (DRA) problem, where $a_{k,t}$ is chosen from a discrete action space. For example, the $a_{k,t}$ may represent the numbers of users that should be allocated to different wireless channels. For simplicity, we assume that the action space of $a_{k,t}$ is $\mathcal{A}_d = \{0, 1, \cdots, N-1\}$ where $|\mathcal{A}_d| = N \leq Q+1$ and Q is again the available budget. Thus, the full allocation space is $\{\boldsymbol{a}_t \mid a_{k,t} \in \mathcal{A}_d, \sum_k a_{k,t} \leq Q\}$. In order to solve the online problem introduced in Section 4.4.2, we adapt a Combinatorial Multi-arm Bandit (CMAB) framework [2]. We maintain a set of base arms $S = \{(k, a) \mid k \in [K], a \in \mathcal{A}_d\}$, where the total number of base arms $|S| = KN$. For each base arm $(k, a) \in S$, we denote the expected reward of playing (k, a) as $\mu_{k,a} = \mathbb{E}_{X_{k,t} \sim D_k}[f_k(a, X_{k,t})]$ and let $\boldsymbol{\mu} = (\mu_{k,a})_{(k,a) \in S}$. We can rewrite the expected total reward obtained in round t as a function of $\boldsymbol{a}_t$ and $\boldsymbol{\mu}$:

$$\begin{aligned} r'(\boldsymbol{a}_t, \boldsymbol{\mu}) &= \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})\right] \\ &= \sum_{k=1}^{K} \sum_{a \in \mathcal{A}_d} \mu_{k,a} \cdot \mathbb{1}\{a_{k,t} = a\}, \end{aligned} \tag{4.9}$$

which reflects the fact that the expected total reward is the sum of the expected reward of all chosen base arms.

Based on the new parameters $\{\mu_{k,a}\}$, we propose the CUCB-DRA solution algorithm described in Alg. 4.4. The algorithm maintains the empirical mean $\hat{\mu}_{k,a}$ and a confidence radius $\rho_{k,a}$ for the reward of each arm $(k, a) \in S$. It feeds the budget Q and all the upper confidence bounds $\{\bar{\mu}_{k,a}\}$ into the offline oracle $\mathcal{O}$ to obtain an allocation $\boldsymbol{a}_t$ for round t. The confidence radius $\rho_{k,a}$ is large if arm (k, a) is not chosen often ($T_{k,a}$, which denotes the number of times this arm has been chosen, is small). We define the reward gap $\Delta_{\boldsymbol{a}} = \max(0, \alpha \cdot \text{opt}(\boldsymbol{D}) - r(\boldsymbol{a}, \boldsymbol{D}))$ for all feasible allocations $\boldsymbol{a} \in \mathcal{A}_d^K, \sum_{k=1}^{K} a_k \leq Q$. For each arm (k, a), we define $\Delta_{\min}^{k,a} = \inf_{\boldsymbol{a} \in \mathcal{A}_d^K : a_k = a, \Delta_{\boldsymbol{a}} > 0} \Delta_{\boldsymbol{a}}$, $\Delta_{\max}^{k,a} = \sup_{\boldsymbol{a} \in \mathcal{A}_d^K : a_k = a, \Delta_{\boldsymbol{a}} > 0} \Delta_{\boldsymbol{a}}$. We define $\Delta_{\min} = \min_{(k,a) \in S} \Delta_{\min}^{k,a}$ and $\Delta_{\max} = \max_{(k,a) \in S} \Delta_{\max}^{k,a}$. We then provide the regret bounds of the CUCB-DRA algorithm.

Theorem 4.6. *For the CUCB-DRA algorithm on an online DRA problem with a bounded smoothness constant $B \in \mathbb{R}^+$ [3]*

Algorithm 4.4 CUCB-DRA with offline oracle $\mathcal{O}$

Input: Budget Q, Oracle $\mathcal{O}$.
For each arm $(k,a) \in S$, $T_{k,a} \leftarrow 0$. {maintain the total number of times arm (k,a) is played so far.}
For each arm $(k,a) \in S$, $\hat{\mu}_{k,a} \leftarrow 0$. {maintain the empirical mean of $f_k(a, X_k)$.}
for $t = 1, 2, 3, \ldots$ **do**
    For each arm $(k,a) \in S, \rho_{k,a} \leftarrow \sqrt{\frac{3 \ln t}{2T_{k,a}}}$. {the confidence radius, $\rho_{k,a} = +\infty$ if $T_{k,a} = 0$.}
    For each arm $(k,a) \in S, \bar{\mu}_{k,a} = \hat{\mu}_{k,a} + \rho_{k,a}$. {the upper confidence bound.}
    $\boldsymbol{a}_t \leftarrow \mathcal{O}((\bar{\mu}_{k,a})_{(k,a)\in S}, Q)$.
    Take allocation $\boldsymbol{a}_t$, observe feedback $f_k(a_{k,t}, X_{k,t})$'s.
    For each $k \in [K]$, update $T_{k,a_{k,t}}$ and $\hat{\mu}_{k,a_{k,t}}$: $T_{k,a_{k,t}} = T_{k,a_{k,t}} + 1, \hat{\mu}_{k,a_{k,t}} = \hat{\mu}_{k,a_{k,t}} + (f_k(a_{k,t}, X_{k,t}) - \hat{\mu}_{k,a_{k,t}})/T_{k,a_{k,t}}$.
end for

1. *if* $\Delta_{\min} > 0$, *we have a distribution-dependent bound*

$$Reg_{\alpha,\beta}(T, \boldsymbol{D}) \leq \sum_{(k,a)\in S} \frac{48B^2 Q \ln T}{\Delta^{k,a}_{\min}} + 2BKN + \frac{\pi^2}{3} \cdot KN \cdot \Delta_{\max} \tag{4.10}$$

2. *we have a distribution-independent bound*

$$Reg_{\alpha,\beta}(T, \boldsymbol{D}) \leq 14B\sqrt{QKNT \ln T} + 2BKN + \frac{\pi^2}{3} \cdot KN \cdot \Delta_{\max}. \tag{4.11}$$

Notice that our regret results hold for any finite discrete action space $\mathcal{A}_d$. The only change in the regrets will be the replacement of N with the actual $|\mathcal{A}_d|$. We rely on the following properties of $r'(\boldsymbol{a}_t, \boldsymbol{\mu})$, which are required by the general CMAB framework in [3], to bound the regret.

Condition 4.1. *(Monotonicity). The reward $r'(\boldsymbol{a}_t, \boldsymbol{\mu})$ satisfies monotonicity, if for any allocation $\boldsymbol{a}_t$, any two vectors $\boldsymbol{\mu} = (\mu_{k,a})_{(k,a)\in S}$, $\boldsymbol{\mu}' = (\mu'_{k,a})_{(k,a)\in S}$, we have $r'(\boldsymbol{a}_t, \boldsymbol{\mu}) \leq r'(\boldsymbol{a}_t, \boldsymbol{\mu}')$, if $\mu_{k,a} \leq \mu'_{k,a}$ for all $(k,a) \in S$.*

Condition 4.2. *(1-Norm Bounded Smoothness). The reward $r'(\boldsymbol{a}_t, \boldsymbol{\mu})$ satisfies the 1-norm bounded smoothness condition, if there exists $B \in \mathbb{R}^+$ (referred as the bounded smoothness constant) such that, for any allocation $\boldsymbol{a}_t$, and any two vectors $\boldsymbol{\mu} = (\mu_{k,a})_{(k,a)\in S}$, $\boldsymbol{\mu}' = (\mu'_{k,a})_{(k,a)\in S}$, we have $|r'(\boldsymbol{a}_t, \boldsymbol{\mu}) - r'(\boldsymbol{a}_t, \boldsymbol{\mu}')| \leq B \sum_{(k,a)\in S} |\mu_{k,a} - \mu'_{k,a}|$.*

It is easy to check that both properties hold for $r'(\boldsymbol{a}_t, \boldsymbol{\mu})$ in Eq. (4.13).

4.4.4 Online Continuous Resource Allocation

In Section 4.4.3, we propose an algorithm to solve the online resource allocation problem with discrete action space $\mathcal{A}$. However, in many real-world resource allocation applications, the budget can be continuous, i.e., $\mathcal{A}$ is an infinite continuous space. In this section, we study the online Continuous Resource Allocation (CRA) problem, where $a_{k,t}$ is chosen from a continuous space $\mathcal{A}_c = [0, Q]$. For example, the actions $a_{k,t}$ may be amounts of electricity that a smart power grid pulls from different electric vehicle charging stations. The full allocation space then becomes $\{\boldsymbol{a}_t \mid a_{k,t} \in \mathcal{A}_c, \sum_k a_{k,t} \leq Q\}$. As in Section 4.4.3's discrete setting, we still define the set of base arms as $S = \{(k, a) \mid k \in [K], a \in \mathcal{A}_c\}$, but unlike the discrete setting, we now have to maintain an infinite number of base arms. Thus, we cannot directly apply the CMAB framework. However, we can use a simple but powerful technique called fixed discretization [69], with the assumption that the reward function $f_k(a_{k,t}, X_{k,t})$ satisfies a Lipschitz condition:

$$|f_k(a, X_{k,t}) - f_k(b, X_{k,t})| \leq L \cdot |a - b|, \tag{4.12}$$

where L is the Lipschitz constant known to the algorithm. This condition is satisfied by many realistic reward functions, e.g., $f_k(a, X_{k,t}) = \max\{a - X_{kt}, 0\}$, which represents a reward of 0 if the allocated budget a does not meet a requirement X_{kt}, with linearly increasing reward otherwise.

We consider a discretization of $\mathcal{A}_c$ and denote it as $\widetilde{\mathcal{A}_c}$. We define $\text{opt}_{\mathcal{A}}(\boldsymbol{D}) = \sup_{a_{k,t} \in \mathcal{A}} r(\boldsymbol{a}_t, \boldsymbol{D})$ as the maximum expected total reward under action space $\mathcal{A}$ and distribution $\boldsymbol{D}$. We can decompose the cumulative regret in Eq. (4.8) as:

$$Reg^{\pi}_{\alpha,\beta}(T; \boldsymbol{D}) = \underbrace{T \cdot \alpha \cdot \beta \cdot \text{opt}_{\widetilde{\mathcal{A}_c}}(\boldsymbol{D}) - \sum_{t=1}^{T} r(\boldsymbol{a}^{\pi}_t, \boldsymbol{D})}_{①} + \underbrace{T \cdot \alpha \cdot \beta \cdot \big(\text{opt}_{\mathcal{A}_c}(\boldsymbol{D}) - \text{opt}_{\widetilde{\mathcal{A}_c}}(\boldsymbol{D})\big)}_{②}, \tag{4.13}$$

where ① is the learning regret under action space $\widetilde{\mathcal{A}_c}$ and ② is the discretization error. Both of them depend on the discretization space $\widetilde{\mathcal{A}_c}$.

We propose a CUCB-CRA algorithm that makes a uniform discretization of the budget for each resource. It divides the original action space $\mathcal{A}_c = [0, Q]$ into

Algorithm 4.5 CUCB-CRA with offline oracle $\mathcal{O}$

1: **Input**: Budget Q, Lipschitz constant L, Time horizon T, Oracle $\mathcal{O}$.
2: Let $\epsilon = (\frac{B^2Q^2 \ln T}{L^2KT})^{\frac{1}{3}}$, discretize $\mathcal{A}_c = [0, Q]$ into $\widetilde{\mathcal{A}_c} = \{0, \epsilon, 2\epsilon, \cdots, (N-1) \cdot \epsilon\}$.
3: Run CUCB-DRA algorithm with discrete action space $\widetilde{\mathcal{A}_c}$.

intervals of fixed length $\epsilon = \frac{Q}{N-1}$, so that $\widetilde{\mathcal{A}_c}$ consists of N multiples of ϵ, i.e., $\widetilde{\mathcal{A}_c} = \{0, \epsilon, 2\epsilon, \cdots, (N-1) \cdot \epsilon\}$. With the Lipschitz condition, it is easy to see that ② $\leq T \cdot \alpha \cdot \beta \cdot LK\epsilon$. To bound ①, we simply view this term as the regret of the discrete resource allocation problem discussed in Section 4.4.3, where the number of base arms is still KN. Based on Theorem 4.6, we know the regret in ① is in the order of $O(B\sqrt{QKNT \ln T})$. Choosing $\epsilon = (\frac{B^2Q^2 \ln T}{L^2KT})^{\frac{1}{3}}$, the regret in Eq.(4.13) is minimized and we have

Theorem 4.7. *For the CUCB-CRA algorithm on an online CRA problem with a bounded smoothness constant B, we have a distribution-independent bound*

$$Reg_{\alpha,\beta}(T, \boldsymbol{D}) \leq O((BQK)^{\frac{2}{3}} L^{\frac{1}{3}} T^{\frac{2}{3}} (\ln T)^{\frac{1}{3}}).$$

We note that Theorem 4.7's distribution-independent regret bound is looser than that of Theorem 4.6 for the discrete resource allocation problem, by a factor of $O\left(T/\ln T\right)^{\frac{1}{6}}$. This factor primarily stems from the additional regret due to the fixed discretization in the continuous case. Adaptive discretization methods, e.g., as proposed in [69, 83] for other MAB problems with continuous arms, may allow further reduction of the regret.

4.4.5 Correlated CMAB for Resource Allocation

Correlated CMAB Framework

In several application settings, there may be some information on the knowledge of reward functions $f_k(a_k, X_k)$. As a result, the knowledge of the reward from one base arm (i, k) may provide some information on the reward that would have been obtained from entity k if budget j was allocated to entity k. This is illustrated in Figure 4.6. The rewards corresponding to a base arm (i, k), i.e., budget i to entity k, is a function of the allocated budget i and underlying randomness X_k associated with entity k. The rewards for base arms (i, k) and (j, k), i.e., different budget allocations

within entity k, are correlated through X_k. There may be also correlations in the rewards across different entities if $X_1, X_2, \ldots X_K$ are correlated. For instance, in the power allocation example, where the objective is to allocate the total power Q among K different channels to maximize the total throughput, the throughput at channel k is given by $\log\left(1 + \frac{a_{k,t}}{X_{k,t}}\right)$. Here, $a_{k,t}$ represents the power allocated in channel k and X_k denotes the hidden noise in channel k at round t. As the expression of throughput, i.e., the reward function $f_k(a_k, X_k)$, is known, the throughput in channel k at power i provides some information on what the reward would have been if power j was allocated to channel k. More generally, rewards obtained from one base arm (i,k) may provide some information on the reward of another base arm (j,ℓ). As a result, the rewards corresponding to different base arms are correlated. We capture the presence of such correlations in the form of *pseudo-rewards*, as defined below:

Definition 4.1 (Pseudo-Reward)**.** *Suppose that we sample the base arm (i,k) and observe reward r. We call a quantity $s_{(j,\ell),(i,k)}(r)$ as the pseudo-reward of base arm (j,ℓ) with respect to base arm (i,k) if it is an upper bound on the conditional expected reward of base arm (j,ℓ), i.e.,*

$$\mathbb{E}[f_\ell(j, X_\ell) \mid f_k(i, X_k) = r \,] \leq s_{(j,\ell),(i,k)}(r). \tag{4.14}$$

For convenience, we set $s_{(j,\ell),(j,\ell)}(r) = r \quad \forall j, \ell$.

When no information is known, pseudo-rewards between two base arms are not known, then they can be set equal to the maximum possible reward. This makes our formulation quite general and in fact subsumes the correlation agnostic combinatorial framework studied in [12]. Next, we show how the pseudo-rewards can be evaluated in practice.

Obtaining pseudo-rewards from reward correlations within the same entity. These pseudo-rewards can be evaluated easily in several different practical settings. For instance, if the form of the functions $f_k(a_k, X_k)$ is known, then the pseudo-reward of base arm (j,k) with respect to base arm (i,k) can be obtained as

$$s_{(j,k),(i,k)}(r) = \max_x f_k(j,x) \quad \text{s.t. } f_k(i,x) = r. \tag{4.15}$$

Note that pseudo-rewards can be obtained even in the scenario where only probabilistic upper and lower bounds on $f_k(a_k, X_k)$ are known, i.e., $\underline{f}_k(a_k, X_k) \leq f_k(a_k, X_k) \leq$

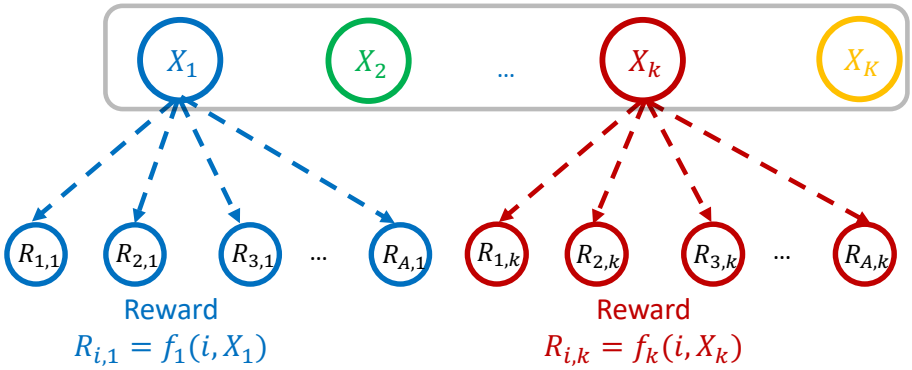


Figure 4.6: Illustration of reward correlation.

$\bar{f}_k(a_k, X_k)$ w.p. $1-\kappa$. In this scenario, we can construct pseudo-rewards as follows:

$$s_{(j,k),(i,k)}(r) = (1-\kappa)^2 \times \left(\max_{\{X_k : \underline{f}_k(i,X_k) \le r \le \bar{f}_k(i,X_k)\}} \bar{f}_k(j, X_k) \right) + (1-(1-\kappa)^2) \times M, \tag{4.16}$$

where M is the maximum possible reward that a base arm can provide. We evaluate this pseudo-reward by first identifying the range of values within which X_k lies based on the reward with probability $1-\kappa$. The maximum possible reward of the base arm (j,k) within the identified range of X_k is then computed with probability $1-\kappa$. Due to this, with probability $(1-\kappa)^2$, the conditional reward of base arm (j,k) is at most $\max_{X_k : \underline{f}_k(i,X_k) \le r \le \bar{f}_k(i,X_k)} \bar{f}_k(j, X_k)$. As the maximum possible reward is M otherwise, we get (4.16).

Obtaining pseudo-rewards from reward correlation across entities. In the most general scenario, there may be knowledge of reward correlations across entities as shown in Figure 4.7. Upon observing a reward r from a base arm, pseudo-rewards $s_{(j,\ell),(i,k)}(r)$, give us an upper bound on the conditional expectation of the reward from base arm (j,ℓ) given that we observed reward r from arm (i,k). The reward received for entity k at a given budget i may provide some information on what the reward would have been if budget j were allocated to entity k, leading to correlations within entity. The rewards of different entities may also be correlated. This can occur if the random variables X_k and X_ℓ, i.e., the hidden random variables corresponding to two different entities k and ℓ, are correlated. These correlations can be incorporated in our framework through pseudo-rewards $s_{(j,\ell),(j,k)}$, which are

an upper bound on the conditional expected reward. For instance, in the application of financial optimization, the company may invest its total budget among different products. As the performance of different products are likely to be correlated, the reward feedback under budget i for product k may inform something about the reward feedback for product ℓ under budget j. Such correlations can be modeled through pseudo-rewards, which may either be known from domain knowledge or from previously performed controlled experiments. For example, based on previously performed experiments, it may be known that the expected reward obtained from product ℓ under budget j is at most y whenever the reward obtained for product k under budget i is x. Note that in this modeling, one does not need to explicitly capture what the inherent randomness X_k represents and its corresponding values. This is a key strength of our proposed framework, as in several applications X_k could be hard to interpret and model. For instance, in the financial optimization example, X_k may represent underlying market conditions that are complex and subsequently the reward functions $f_k(a_k, k)$ are also unknown. Even in such settings, the pseudo-reward based framework allows one to capture the correlation across different base arms.

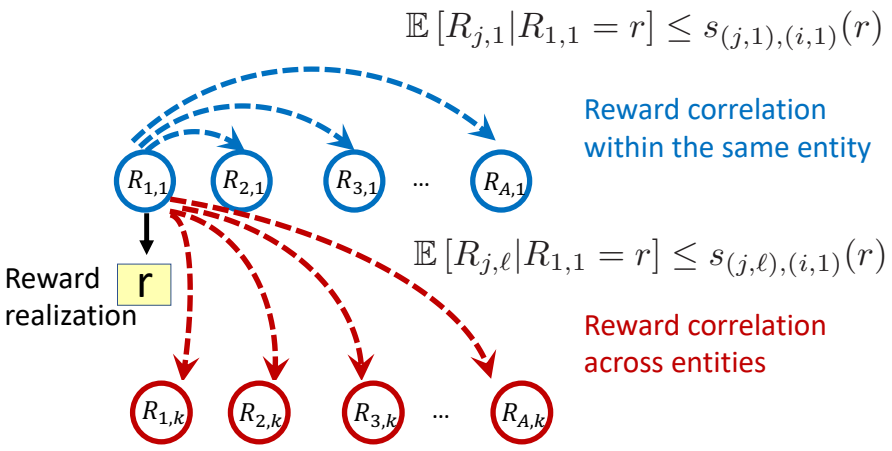


Figure 4.7: Pseudo-rewards from reward correlation across entities.

Correlated-UCB Algorithm

We now propose the correlated-Upper Confidence Bound algorithm for resource allocation (corr-UCB-RA) that uses existing correlation in rewards across base arms to maximize the long-term cumulative reward.

Algorithm 4.6 Correlated UCB for resource allocation with offline oracle $\mathcal{O}$

Input: Oracle $\mathcal{O}$.
For each base arm $(i,k) \in \mathcal{A} \times \mathcal{K}$, $n_{i,k}(t) \leftarrow 0$. {maintain the total number of times base arm (i,k) is played so far.}
for $t = 1, 2, 3, \ldots$ **do**
  For each base arm $(j,\ell) \in \mathcal{K} \times \mathcal{A}$, evaluate its KA pseudoUCB indices $U_{(j,\ell),(i,k)}(t) \triangleq \hat{\phi}_{(j,\ell),(i,k)}(t) + B\sqrt{\frac{2\log t}{n_{(i,k)}(t)}}$
  For each $(j,\ell) \in \mathcal{A} \times \mathcal{K}$, $U_{(j,\ell)}(t) = \min_{(i,k)} U_{(j,\ell),(i,k)}(t)$
  $\boldsymbol{S_t} \leftarrow \mathcal{O}((U_{i,k}(t))_{(i,k)\in\mathcal{A}\times\mathcal{K}})$
  Take allocation $\boldsymbol{S}_t$, observe feedback $f_k(a_{k,t}, X_{k,t})$'s
  Update $n_{(a_{k,t},k)}$, the empirical pseudo-rewards $\hat{\phi}_{(j,\ell),(i,k)}(t)$ for all (j,ℓ), the empirical reward for base arm $(i,k) \in \mathcal{S}_t$
end for

Under the correlated combinatorial bandit framework, the pseudo-reward for base arm (j,ℓ) with respect to the base arm (i,k) provides an estimate on the reward of base arm (j,ℓ) based on the reward obtained from base arm (i,k). We now define the notion of empirical pseudo-reward, which can be used to obtain an *optimistic estimate* of $\mu_{(j,\ell)}$ through just reward samples of base arm (i,k).

Definition 4.2 (Empirical and Expected Pseudo-Reward)**.** *After t rounds, a base arm (i,k) is sampled $n_{(i,k)}(t)$ times. Using these $n_{(i,k)}(t)$ reward realizations, we can construct the empirical pseudo-reward $\hat{\phi}_{(j,\ell),(i,k)}(t)$ for each base arm (j,ℓ) with respect to base arm (i,k) as follows.*

$$\hat{\phi}_{(j,\ell),(i,k)}(t) \triangleq \frac{\sum_{\tau=1}^{t} \mathbb{1}_{(i,k)\in\mathcal{S}_t}\ s_{(j,\ell),(i,k)}(f_k(i, X_{k,\tau}))}{n_{(i,k)}(t)}, \tag{4.17}$$

$$(j,\ell) \in \mathcal{K} \times \mathcal{A} \setminus \{(i,k)\}. \tag{4.18}$$

The expected pseudo-reward of base arm (j,ℓ) with respect to base arm (i,k) is defined as

$$\phi_{(j,\ell),(i,k)} \triangleq \mathbb{E}s_{(j,\ell),(i,k)}(f_k(i, X_k)). \tag{4.19}$$

For convenience, we set $\hat{\phi}_{(i,k),(i,k)}(t) = \hat{\mu}_{(i,k)}(t)$ and $\phi_{(i,k),(i,k)} = \mu_{(i,k)}$. Note that the empirical pseudo-reward $\hat{\phi}_{(j,\ell),(i,k)}(t)$ is defined with respect to base arm (i,k) and it is only a function of the rewards observed by sampling base arm (i,k).

Definition 4.3 (PseudoUCB Index $U_{(j,\ell),(i,k)}(t)$)**.** *We define the PseudoUCB Index of base arm (j,ℓ) with respect to base arm (i,k) as follows.*

$$U_{(j,\ell),(i,k)}(t) \triangleq \hat{\phi}_{(j,\ell),(i,k)}(t) + \sqrt{\frac{2\log t}{n_{(i,k)}(t)}}. \tag{4.20}$$

Furthermore, we define $U_{(j,\ell)}(t) = \min_{(i,k)} U_{(j,\ell),(i,k)}(t)$, the tightest of the KA upper bounds for base arm (j,ℓ).

At each round, the algorithm computes these pseudo-UCB indices $U_{(j,\ell)}$ for each base arm (j,ℓ). These indices are then fed to the oracle to obtain the budget allocation vector $\boldsymbol{S_t}$ at round t. At the end of each round we update the empirical pseudo-rewards $\hat{\phi}_{(j,\ell),(i,k)}(t)$ for all (j,ℓ), the empirical reward for arm $(i,k) \in \mathcal{S}_t$, where $\mathcal{S}_t$ denotes the set of base arms played in round t. We next show the regret bound of the proposed Corr-UCB-RA algorithm. To state our result, we first define the notion of competitive and non-competitive base arms.

Definition 4.4 (Competitive and Non-Competitive base arms)**.** *If $\phi_{(j,\ell),(i,k)} \leq \bar{\mu}_{(j,\ell)}$ for some $(i,k) \in \mathcal{S}^*$ then base arm (j,ℓ) is called Non-competitive, otherwise it is called Competitive. Here, $\mathcal{S}^*$ denotes the set of base arms played in the oracle's optimal budget allocation vector $\boldsymbol{S^*}$. Furthermore, we define the pseudo-gap of a base arm (j,ℓ) as $\bar{\Delta}_{(j,\ell)} = \bar{\mu}_{(j,\ell)} - \max_{(i,k)\in\mathcal{S}^*} \phi_{(j,\ell),(i,k)}$.*

Note that the pseudo-gap is greater than zero for non-competitive base arms and is less than or equal to zero for competitive base arms. The definition of pseudo-gap is useful to state our regret bounds. Intuitively, a base arm (j,ℓ) is non-competitive if it can be inferred that the mean reward of (j,ℓ) is smaller than the threshold $\bar{\mu}_{(j,\ell)}$ through just the samples of a base arm belonging to the oracle's optimal budget allocation $\mathcal{S}^*$. In what follows, we refer to the total number of competitive base arms as C and the set of competitive base arms as $\mathcal{C}$. As mentioned earlier, the Corr-UCB-RA algorithm selects the budget allocation $\boldsymbol{S}^*$ with high probability if the indices of base arms $U_{(i,k)}$ are *close* to their true means. In the presence of correlations, we show that this can be achieved by sampling competitive base arms $O(\log T)$ times and non-competitive base arms only $O(1)$ times. This occurs as the non-competitive base arms can be identified as sub-optimal based on samples of optimal base arms. We formalize this intuition to get the following regret bound for our Corr-UCB algorithm.

Theorem 4.8. *The expected cumulative regret of the Correlated-UCB algorithm for resource allocation is upper bounded as*

$$Reg_{\alpha}(T, \boldsymbol{D}) \leq \sum_{(i,k)\in\mathcal{C}} \Delta_{max}^{(i,k)} \left(\frac{8\log T}{\left(g^{-1}\left(\Delta_{min}^{(i,k)}\right)\right)^2} + 2 \right) \quad + \sum_{(i',k')\in\mathcal{K}\times\mathcal{A}\setminus\{\mathcal{C}\}} \Delta_{max}^{(i',k')}(4KAt_0 + 6(KA)^3) + 2(KA)^2\Delta_{max} \tag{4.21}$$

$$= C \cdot O(\log T) + O(1), \tag{4.22}$$

where $\mathcal{C} \subseteq \mathcal{K}\times\mathcal{A}$ *is set of competitive base arms with cardinality* C*, and* $t_0 = \inf\Bigg\{\tau \geq 2 : g^{-1}\left(\Delta_{min}^{(i,k)}\right) \geq 4\sqrt{\frac{2K\log\tau}{\tau}} \quad \forall(i,k), \bar{\Delta}_{(i,k)} \geq 4\sqrt{\frac{2K\log\tau}{\tau}} \quad \forall(i,k) \in \mathcal{A}\times\mathcal{K}\setminus\mathcal{C}\Bigg\}$.

Remark 1 (Competitive and Non-competitive base arms). Recall that a base arm (i,k) is said to be non-competitive if the expected pseudo-reward of base arm (i,k) with respect to some base $(j,\ell) \in \mathcal{S}^*$ is smaller than $\bar{\mu}_{(i,k)}$. Note that the optimal set of arms $\mathcal{S}^*$, reward distributions of individual base arms are unknown at the beginning and as a result, the Corr-UCB-RA initially does not know which base arms are competitive and non-competitive.

Remark 2 (Reduction in the effective set of base arms). Upon comparison with the regret of the UCB-RA algorithm, from Lemma 1, we see that our proposed algorithm reduces the regret from $KA \times O(\log T)$ to $C \times O(\log T)$, since only C out of the total KA need to be sampled $O(\log T)$ times before the condition in Claim 1 is met with high probability. As a result, the Corr-UCB-RA only explores C out of the KA base arms explicitly and effectively reduces the problem with KA base arms to one with C base arms.

Remark 3 (Bounded regret in certain settings). Whenever the set $\mathcal{C}$ is empty, the proposed Corr-UCB-RA algorithm achieves bounded regret, which is an order-wise improvement over the regret of correlation agnostic UCB-RA algorithm. One scenario in which this can occur is if the functions $f_k(\cdot)$ are invertible with respect to X_k given a_k. More generally, whenever the sub-optimal base arms can be identified as sub-optimal through just the samples of optimal base arms, we get a bounded regret. Note that the algorithm initially has no knowledge about the optimality/sub-optimality of base arms and in such cases it identifies them by

sampling the sub-optimal base arms only $O(1)$ times.

4.4.6 Experiments

To validate our theoretical findings, we conduct experiments on three applications in wireless networks. First, we apply our CUCB-DRA algorithm to the dynamic channel allocation problem, where multiple wireless devices share a limited number of wireless channels. Second, we use the CUCB-CRA algorithm to solve the online water filling problem [84], which is essential to the power allocation in OFDM systems [85]. Finally, we use the Corr-UCB algorithm to solve a dynamic user allocation problem in wireless networks, where we need to allocate new incoming users to different wireless access points with an unknown number of existing users. We evaluate our algorithm in the setting with a non-invertible reward function.

Dynamic Channel Allocation

For the dynamic channel allocation problem, we set $K = 4, Q = 8$, which represents that there are 8 wireless devices sharing 4 orthogonal channels. We assume that each device uses the slotted-ALOHA protocol with the same traffic load 0.05, and each channel has random background traffic uniformly sampled from [0.05, 0.15]. Our goal is to allocate one channel to each device such that their total throughput is maximized. In this setting, we use the CUCB-DRA algorithm, where the base arm (k, a) becomes assigning a devices to channel k, and the expected reward f_k is the average throughput of channel k. We repeat the experiment 50 times, and Figure 4.8 shows the average regrets of different methods with 95% confidence interval. We shrink the confidence radius in our CUCB-DRA algorithm by γ, i.e., $\rho_{k,a} \leftarrow \gamma\sqrt{3 \ln t / 2T_{k,a}}$, to speed up the learning, though our theoretical regret bound requires $\gamma = 1$. We compare CUCB-DRA to the ϵ-greedy algorithm with exploration probability $\epsilon = 0.01$ and find CUCB-DRA with $\gamma = 0.01$ achieves 50% less regret than ϵ-greedy over 5000 rounds.

Online Water Filling

To evaluate the performance of CUCB-CRA, we consider the water filling problem where a total amount of one unit of power has to be assigned to 3 communication

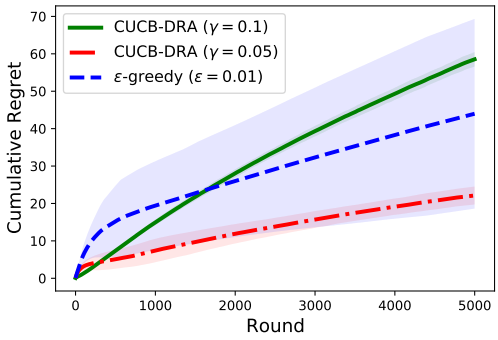


Figure 4.8: Regrets of CUCB-DRA for dynamic channel allocation problem.

channels, i.e., $Q = 1, K = 3$, with the objective of maximizing the total communication rate. The communication rate of the i^{th} channel is given by $\log(\beta_i + x_i)$, where x_i represents the power allocated to channel i and β_i represents the floor above the baseline at which power can be added to the channel. It can be written as a convex optimization problem:

$$\begin{aligned} \underset{x_i}{\text{maximize}} \quad & \sum_{i=1}^{K} \log(\beta_i + x_i) \\ \text{subject to} \quad & \sum_{i=1}^{K} x_i = Q, x_i \geq 0. \end{aligned} \tag{4.23}$$

For the online water filling problem, the $\{\beta_i\}$ are unknown and uniformly sampled from $[0.8, 1.2]$. As it is a online continuous resource allocation problem, we choose different discretization granularities ϵ from $\{0.25, 0.2, 0.1.0.05\}$ for CUCB-CRA, i.e., $N = 4, 5, 10, 20$. We repeat the experiment 50 times and Figure 4.9 shows the average regrets with 95% confidence interval. The results of CUCB-CRA are consistent with our analysis in Eq.(4.13): with the increase of the discretization granularity ϵ, the learning regret decreases and the discretization error increases; as a result, the overall regret first decreases then increases.

Dynamic User Allocation

In this section, we apply our corr-UCB-RA algorithm to a dynamic user allocation problem in wireless networks. Our goal is maximize the total throughput of wireless

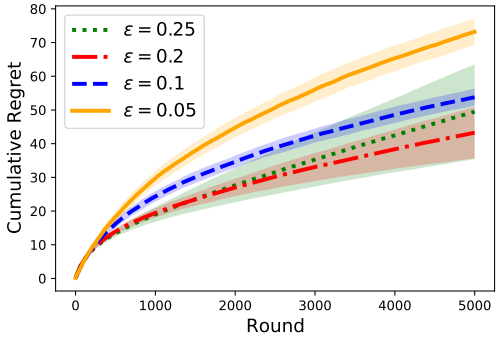


Figure 4.9: Regrets of CUCB-CRA for the online water filling problem.

access points (APs) by allocating new incoming users to them. The number of existing users associated to each AP is time-varying, which affects the traffic load on the AP. We assume each user has a fixed traffic load of 0.2 and consider the well-known ALOHA protocol [86] for each AP. We consider K APs and Q new incoming users at each round. Let X_k denote the number of existing users in each AP k and a_k denote the number of new users allocated to it. Note that we assume all the users of an AP will leave when the round ends, so a_k in the current round will not affect X_k in the future rounds. Our goal is to maximize the total throughput:

$$\begin{aligned} \underset{a_k}{\text{maximize}} \quad & \sum_{k=1}^{K} 0.2(X_k + a_k)e^{-0.2(X_k+a_k)} \\ \text{subject to} \quad & \sum_{i=k}^{K} a_k = Q, a_k \in \mathbb{N}. \end{aligned}$$

We extract $\{X_k\}$, the number of existing users in each AP, from a real-world dataset [87]. We choose 4 APs (91, 92, 94, 95) on the 3rd floor of Building 3 on campus, and record their associated users from 13:00 to 16:00 on March 2, 2015. The detailed distribution of the number of existing users on different access points can be found in the Appendix. In our experiment, at each round, we first sample $\{X_k\}$ from the extracted distribution, then allocate $Q = 8$ new users to these four APs. Since the throughput function is non-invertible, our algorithm cannot directly infer X_k from the observed throughput of each AP and needs to maintain the pseudoUCB indices of base arms. We compare it with the UCB-RA algorithm. Figure 4.11 shows the average regrets with 95% confidence interval over 20 experiments. The result is

consistent with our analysis: corr-UCB-RA achieves 25% less regret than correlation agnostic UCB-RA algorithm. This occurs as the corr-UCB-RA algorithm is able to make use of the correlations between the reward of base arms to incur a regret of $C \cdot O(\log T)$ as opposed to $KA \cdot O(\log T)$. We also show the relationship between Q and the total regret after 2000 rounds in Figure 4.10: with the increase of Q, the total regret of corr-UCB-RA increases much more slowly than that of UCB-RA.

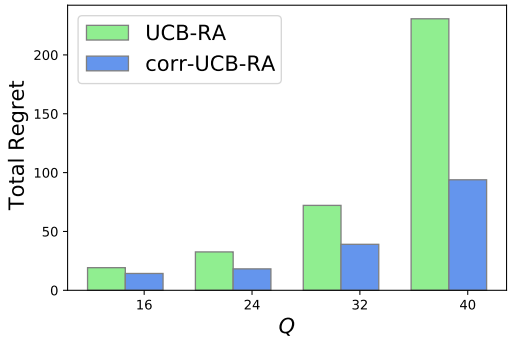


Figure 4.10: Regret comparison between CUCB-CRA and Corr-UCB-RA.

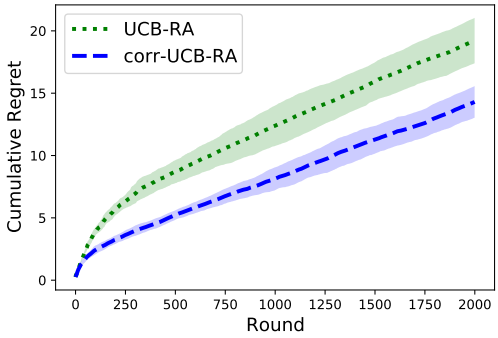


Figure 4.11: Performance comparison between Corr-UCB-RA and CUCB-CRA.

4.5 Summary

We introduce the centralized and distributed versions of the competitive CMAB problem from the multi-players' perspective in this chapter. We discuss both settings in the dynamic channel allocation application and propose algorithms with regret guarantees. We also study the general online resource allocation problem with the centralized setting and discuss how the correlated information can help improve the regret bounds of the proposed algorithms.

4.6 Proof

4.6.1 Proof of Lemma 4.1, 4.2

Proof. Assume there exists an observation list $\boldsymbol{o}_{\text{old}}$ such that $o_{\text{old}}^{(i)} = b, o_{\text{old}}^{(j)} = a$, and $i < j, \mu_a > \mu_b$. In other words, the i^{th} arm to be observed in $\boldsymbol{o}_{\text{old}}$ has less expected reward than the j^{th} arm. Now let us consider a new observation list $\boldsymbol{o}_{\text{new}}$, which switches arms a and b in $\boldsymbol{o}_{\text{old}}$ and leaves the other arms unchanged. Define the one-round expected reward of $\boldsymbol{o}_{\text{old}}$ and $\boldsymbol{o}_{\text{new}}$ as r_{old} and r_{new}. From (4.3), we can find that the gap between r_{old} and r_{new} is only caused by the i^{th} to the j^{th} arm in the observation list, so we get:

$$
\begin{aligned}
&r_{\text{new}} - r_{\text{old}} = \sum_{k=i}^{j} \left\{ (1-k\,\tau)\mu_{o_{\text{new}}^{(k)}} \prod_{x=1}^{k-1}(1-\mu_{o_{\text{new}}^{(x)}}) - (1-k\,\tau)\mu_{o_{\text{old}}^{(k)}} \prod_{x=1}^{k-1}(1-\mu_{o_{\text{old}}^{(x)}}) \right\} \\
&= \prod_{x=1}^{i-1}(1-\mu_{o_{\text{new}}^{(x)}}) \Bigg\{ (1-i\,\tau)(\mu_a-\mu_b) - \sum_{k=i+1}^{j-1} \Big\{ (1-k\,\tau)(\mu_a-\mu_b)\mu_{o_{\text{new}}^{(k)}} \prod_{x=i+1}^{k-1}(1-\mu_{o_{\text{new}}^{(x)}}) \Big\} - \\
&(1-j\,\tau)(\mu_a-\mu_b) \prod_{x=i+1}^{j-1}(1-\mu_{o_{\text{new}}^{(x)}}) \Bigg\} \\
&> \prod_{x=1}^{i-1}(1-\mu_{o_{\text{new}}^{(x)}}) \Bigg\{ (1-i\,\tau)(\mu_a-\mu_b) - (1-(i+1)\,\tau)(\mu_a-\mu_b) \sum_{k=i+1}^{j} \mu_{o_{\text{new}}^{(k)}} \prod_{x=i+1}^{k-1}(1-\mu_{o_{\text{new}}^{(x)}}) - \\
&(1-(i+1)\,\tau)(\mu_a-\mu_b) \prod_{x=i+1}^{j-1}(1-\mu_{o_{\text{new}}^{(x)}}) \Bigg\} \\
&\geq \prod_{x=1}^{i-1}(1-\mu_{o_{\text{new}}^{(x)}}) \Big\{ (1-i\,\tau)(\mu_a-\mu_b) - (1-(i+1)\,\tau)(\mu_a-\mu_b) \Big\} \\
&\geq \prod_{x=1}^{i-1}(1-\mu_{o_{\text{new}}^{(x)}}) \Big\{ \tau\,(\mu_a-\mu_b) \Big\} \\
&\geq 0.
\end{aligned}
\tag{4.24}
$$

Thus, the expected reward of $\boldsymbol{o}_{\text{new}}$ is always larger than that of $\boldsymbol{o}_{\text{old}}$. As a result, exchanging arms a and b in $\boldsymbol{o}_{\text{old}}$ always improves the expected reward. We can then conclude that the optimal policy for the single-player setting is $\boldsymbol{o}_t^* = (1, 2, \ldots, K)$, which is Lemma 4.1. For the centralized multi-player setting, similarly, the optimal ordering is where no arm has lower expected reward than any arm observed after it,

which concludes the proof of Lemma 4.2. □

4.6.2 Proof of Theorem 4.1

Proof. To prove Theorem 4.1, let us firstly rewrite (4.4) as:

$$\begin{aligned} R(T) &= \sum_{t=1}^{T}\sum_{k=1}^{K}\left\{(1-k\,\tau)\mu_k\prod_{i=1}^{k-1}(1-\mu_i) - (1-k\,\tau)\mu_{o_t^{(k)}}\prod_{i=1}^{k-1}(1-\mu_{o_t^{(i)}})\right\} \\ &\leq \sum_{t=1}^{T}\sum_{k=1}^{K}\left\{(1-k\,\tau)(\mu_k-\mu_{o_t^{(k)}})\prod_{i=1}^{k-1}(1-\mu_i)\right\}. \end{aligned} \tag{4.25}$$

The last inequality holds since $\prod_{i=1}^{k-1}(1-\mu_i)$ is always not greater than $\prod_{i=1}^{k-1}(1-\mu_{o_t^{(i)}})$ for any $\boldsymbol{o}_t$ when $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$. Now let us focus on this inequality. At round t, if $o_t^{(k)} > k$ (i.e., the kth pre-observed arm has better average reward than arm k), then $\mu_k - \mu_{o_t^{(k)}} \geq 0$ and the regret for $o_t^{(k)}$ is nonnegative; if $o_t^{(k)} < k$, then $\mu_k - \mu_{o_t^{(k)}} \leq 0$, and the regret for $o_t^{(k)}$ is non-positive. In order to upper bound $R(T)$, we can ignore the negative terms and only count the positive regrets for all $o_t^{(k)} > k$. These positive regrets come from observing arms with lower expected rewards before those with higher expected rewards. Letting $W_k := (1-k\,\tau)\prod_{i=1}^{k-1}(1-\mu_i)$ and $\Delta_{i,j} := \mu_i - \mu_j$, the total regret can be bounded as:

$$R(T) \leq \sum_{t=1}^{T}\sum_{k=1}^{K}\left\{W_k\,\Delta_{k,o_t^{(k)}}\mathbb{1}\{o_t^{(k)} > k\}\right\}. \tag{4.26}$$

Define $T_{i,j}$ as the number of times that the i^{th} arm to be observed in $\boldsymbol{o}_t$ is arm j, i.e., $T_{i,j} := \sum_{t=1}^{T}\mathbb{1}\{o_t^{(i)} = j\}$. We then rewrite (4.26):

$$\begin{aligned} R(T) &\leq \sum_{t=1}^{T}\sum_{i=1}^{K-1}\sum_{j=i+1}^{K}\left\{W_i\,\Delta_{i,j}\mathbb{1}\{o_t^{(i)} = j\}\right\} \\ &= \sum_{i=1}^{K-1}\sum_{j=i+1}^{K}\left\{W_i\,\Delta_{i,j}\sum_{t=1}^{T}\mathbb{1}\{o_t^{(i)} = j\}\right\} \\ &= \sum_{i=1}^{K-1}\sum_{j=i+1}^{K}\left\{W_i\,\Delta_{i,j}\,T_{i,j}\right\}. \end{aligned} \tag{4.27}$$

In order to bound $\mathbb{E}[R(T)]$, we need to bound $\mathbb{E}[T_{i,j}]$ for all $i < j$.

Lemma 4.4. *$\forall\, i, j \in [K]$ with $i < j$, under Algorithm 1, $\mathbb{E}[T_{i,j}] \leq i\,(\frac{8\log T}{\Delta_{i,j}^2} + 1 + \frac{\pi^2}{3})$.*

Proof. Algorithm 1 sorts the UCB values to determine the pre-observation list $\boldsymbol{o}_t$, so $T_{i,j}$ is equal to the number of times that $\hat{\mu}_j(t)$, the UCB value of arm j, is the i^{th} largest one in $\hat{\boldsymbol{\mu}}(t)$. In that case, at least one arm in the set $\{1, 2, \ldots, i\}$ has smaller UCB value than $\hat{\mu}_j(t)$, since at most $i-1$ arms have larger UCB values than $\hat{\mu}_j(t)$. Thus, $T_{i,j}$ can be bounded by the number of times that the minimum UCB value of arms $\{1, 2, \ldots, i\}$ is less than $\hat{\mu}_j(t)$:

$$\begin{aligned} T_{i,j} &\leq \sum_{t=1}^{T} \mathbb{1}\{\min_{k\in[i]} \hat{\mu}_k(t) \leq \hat{\mu}_j(t)\} \\ &\leq \sum_{t=1}^{T}\sum_{k=1}^{i} \mathbb{1}\{\hat{\mu}_k(t) \leq \hat{\mu}_j(t)\} \\ &\leq \sum_{k=1}^{i}\sum_{t=1}^{T} \mathbb{1}\{\hat{\mu}_k(t) \leq \hat{\mu}_j(t)\}. \end{aligned} \tag{4.28}$$

Since $i < j$ and $k \in [i]$, we can bound $\sum_{t=1}^{T} \mathbb{1}\{\hat{\mu}_k(t) \leq \hat{\mu}_j(t)\}$ using the same idea to bound the number of times of choosing sub-optimal arms in traditional UCB1 algorithm [55]. We can get:

$$\begin{aligned} \mathbb{E}[T_{i,j}] &\leq \sum_{k=1}^{i} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\hat{\mu}_k(t) \leq \hat{\mu}_j(t)\}\right] \\ &\leq \sum_{k=1}^{i} \left\{\frac{8\log T}{\Delta_{k,j}^2} + 1 + \frac{\pi^2}{3}\right\} \\ &\leq i\,(\frac{8\log T}{\Delta_{i,j}^2} + 1 + \frac{\pi^2}{3}), \end{aligned} \tag{4.29}$$

which concludes the proof. □

Combining Lemma 4.4 and (4.27) gives the upper bound of the expected regret

in Theorem 4.1:

$$\begin{aligned}\mathbb{E}[R(T)] &\leq \sum_{i=1}^{K-1}\sum_{j=i+1}^{K}\left\{W_i\,\Delta_{i,j}\mathbb{E}[T_{i,j}]\right\} \\ &\leq \sum_{i=1}^{K-1}\left\{i\,W_i\sum_{j=i+1}^{K}[\frac{8\log T}{\Delta_{i,j}}+(1+\frac{\pi^2}{3})\Delta_{i,j}]\right\}.\end{aligned} \tag{4.30}$$

□

4.6.3 Proof of Lemma 4.3

Proof. When $K \leq 2M$, there are at most two observation steps for each player. As shown in Figure 4.2b, we assume μ_a, μ_b is larger than μ_c, μ_d, and now the expected reward for player 1 and player 2 is $r_{\text{old}} = (1-\tau)(\mu_a+\mu_b) + (1-2\tau)[(1-\mu_a)\mu_c + (1-\mu_b)\mu_d]$. If we switch arms with μ_a and μ_d, the expected reward becomes $r_{\text{new}} = (1-\tau)(\mu_d+\mu_b) + (1-2\tau)[(1-\mu_d)\mu_c + (1-\mu_b)\mu_a]$, so the gap between them is:

$$\begin{aligned}r_{\text{old}} - r_{\text{new}} &= (1-\tau)(\mu_a-\mu_d) - (1-2\tau)(1-\mu_b+\mu_c)(\mu_a-\mu_d) \\ &\geq (1-\tau)(\mu_a-\mu_d) - (1-2\tau)(\mu_a-\mu_d) \\ &\geq \tau(\mu_a-\mu_d) \\ &\geq 0.\end{aligned} \tag{4.31}$$

So the expected reward will only decrease when switching an arm with lower expected reward from step 2 to step 1, which ensures the optimal offline policy to be a greedy policy. □

4.6.4 Proof of Theorem 4.3

Proof. Unlike (4.25), we cannot directly upper bound (4.6) since $\prod_{i=1}^{k-1}(1-\mu_{(k-1)M+m})$ is not always less than $\prod_{i=1}^{k-1}(1-\mu_{o_{m,t}^{(i)}})$. Due to the correlation between different players' expected rewards, the analysis of the regret is challenging. Our idea is to decompose the regret into two parts: the first part is the regret caused by putting the arms into the wrong observation steps; the second part is the regret caused by different arm allocations within one observation step, where the set of arms to be allocated is correct. Define $R_{i,k}^{s}(T)$ as the regret caused by putting arm $i > kM$

into a wrong observation step k, when all previous observation steps are correct. In Figure 4.2b's illustration, this corresponds to an arm being placed in the incorrect column, though the arms in prior columns are placed correctly. We will show why this is sufficient to capture the first part of the total regret. Define $R^a_{i,k}$ as the regret caused by arm i in the correct observation step k, i.e., $(k-1)M+1 \leq i \leq kM$, to capture the second part of the total regret. This regret corresponds to arm i being placed in the correct column k but incorrect row in Figure 4.2b. We can then rewrite the total regret as:

$$R(T) \leq \sum_{k=1}^{L} \left\{ \sum_{i>kM}^{K} R^s_{i,k}(T) + \sum_{i=(k-1)M+1}^{kM} R^a_{i,k}(T) \right\}. \tag{4.32}$$

In order to find the upper bound of $R(T)$, we need to bound $R^s_{i,k}(T)$ and $R^a_{i,k}(T)$ separately. Let us first consider $R^s_{i,k}(T)$. Denote $T^s_{i,k}$ as the number of times that arm i is in the k^{th} observation step. Under algorithm 4.2, we can bound $\mathbb{E}[T^s_{i,k}]$ for all $i > kM$.

Lemma 4.5. *We have* $\mathbb{E}[T^s_{i,k}] \leq kM\,(\frac{8\log T}{\Delta^2_{kM,i}} + 1 + \frac{\pi^2}{3}), \forall\ i > kM.$

Proof. Algorithm 2 sorts the UCB values to determine $\boldsymbol{o}_{m,t}$, so $T^s_{i,k}$ is equal to the number of times that $\hat{\mu}_i(t)$, the UCB value of arm i, should be at least the kM^{th} largest one in $\hat{\boldsymbol{\mu}}(t)$. In that case, at least one arm in the set $\{1, 2, \ldots, kM\}$ has smaller UCB value than $\hat{\mu}_i(t)$, since at most $kM-1$ arms have larger UCB values than $\hat{\mu}_i(t)$. Thus, $T^s_{i,k}$ can be bounded by the number of times that the minimum UCB value of arms $\{1, 2, \ldots, kM\}$ is less than $\hat{\mu}_i(t)$:

$$\begin{aligned} T^s_{i,k} &\leq \sum_{t=1}^{T} \mathbb{1}\{\min_{j\in[kM]} \hat{\mu}_j(t) \leq \hat{\mu}_i(t)\} \\ &\leq \sum_{t=1}^{T}\sum_{j=1}^{kM} \mathbb{1}\{\hat{\mu}_j(t) \leq \hat{\mu}_i(t)\} \\ &\leq \sum_{j=1}^{kM}\sum_{t=1}^{T} \mathbb{1}\{\hat{\mu}_j(t) \leq \hat{\mu}_i(t)\}. \end{aligned} \tag{4.33}$$

Since $i > kM$ and $j \in [kM]$, we can bound $\sum_{t=1}^{T} \mathbb{1}\{\hat{\mu}_j(t) \leq \hat{\mu}_i(t)\}$ using the same idea to bound the number of times of choosing sub-optimal arms in traditional UCB1

algorithm [55]. We can get:

$$\begin{aligned}\mathbb{E}[T^s_{i,k}] &\leq \sum_{j=1}^{kM} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\hat{\mu}_j(t) \leq \hat{\mu}_i(t)\}\right] \\ &\leq \sum_{j=1}^{kM} \left\{\frac{8\log T}{\Delta^2_{j,i}} + 1 + \frac{\pi^2}{3}\right\} \\ &\leq kM\,(\frac{8\log T}{\Delta^2_{kM,i}} + 1 + \frac{\pi^2}{3}),\end{aligned} \tag{4.34}$$

which concludes the proof. □

In order to find the upper bound of $R^s_{i,k}(T)$, we also need to consider the value of regret in each round. Define $R^{\max}_k$ as the maximum one-round regret for one player when he has a wrong arm in the k^{th} observation step and all previous selected arms are correct. We consider the worst case to get this maximum regret, which puts this wrong arm i on the first place in the k^{th} observation step, i.e., $o^{(k)}_{1,t} = i$, since $\mu_{1+(k-1)M} > \mu_{2+(k-1)M} > \cdots > \mu_{kM}$. From (4.5), we can get:

$$\begin{aligned}R^{\max}_k &\leq \sum_{j=k}^{L} \left\{(1-j\tau)\mu_{(j-1)M+1} \prod_{x=1}^{k-1}(1-\mu_{(x-1)M+1})\right\} \\ &\leq (L-k+1)\,\mu_{(k-1)M+1}.\end{aligned} \tag{4.35}$$

Recall that $\alpha = \frac{\mu_{\max}}{\Delta_{\min}}$, where $\mu_{\max} = \max_i \mu_i$ and $\Delta_{\min} = \min_{i<j} \Delta_{i,j}$. Combining Lemma 4.5 and (4.35) gives the upper bound of $R^s_{i,k}(T)$:

$$\begin{aligned}\mathbb{E}[R^s_{i,k}(T)] &\leq R^{\max}_k\, \mathbb{E}[T^s_{i,k}] \\ &\leq kM(L-k+1)(\frac{8\log T}{\Delta^2_{kM,i}} + 1 + \frac{\pi^2}{3})\mu_{1+(j-1)M} \\ &\leq \alpha kM(L-k+1)\left[\frac{8\log T}{\Delta_{kM,i}} + (1+\frac{\pi^2}{3})\Delta_{kM,i}\right]\end{aligned} \tag{4.36}$$

Let us move to the discussion of $R^a_{i,k}(T)$. This part of the regret comes from the fact that, at the k^{th} observation step, although players choose from the correct set of arms $\{(k-1)M+1, (k-1)M+2, \ldots, kM\}$, there are $M!$ possible allocations, which might cause regret compared to the baseline policy. Now we need to consider

the regret of putting arm i into the wrong place within the correct observation step k, where $(k-1)M+1 \leq i \leq kM$. Denote $T^a_{i,k}$ as the number of times that arm i appears in a wrong place at the correct observation step k. Under Algorithm 4.2, we can bound $\mathbb{E}[T^a_{i,k}]$ for all $(k-1)M+1 \leq i \leq kM$.

Lemma 4.6. *For all $(k-1)M+1 \leq i \leq kM$, under Algorithm 4.2, $\mathbb{E}[T^a_{i,k}] \leq (i-1)\,(\frac{8\log T}{\Delta^2_{i-1,i}}+1+\frac{\pi^2}{3})+(K-i)\,(\frac{8\log T}{\Delta^2_{i,i+1}}+1+\frac{\pi^2}{3})$.*

Proof. Let us first consider that arm i appears before its correct place and denote the number of times it happens as $T^{a-}_{i,k}$. Algorithm 4.2 sorts the UCB values of arms, so $T^{a-}_{i,k}$ is equal to the number of times that $\hat{\mu}_i(t)$, the UCB value of arm i, is at least the $i-1$ largest one in $\hat{\boldsymbol{\mu}}(t)$. In that case, at least one arm in the set $\{1,2,\ldots,i-1\}$ has smaller UCB value than $\hat{\mu}_i(t)$, since at most $i-2$ arms have larger UCB values than $\hat{\mu}_i(t)$. Thus, $T^{a-}_{i,k}$ can be bounded by the number of times that the minimum UCB value of arms $\{1,2,\ldots,i-1\}$ is less than $\hat{\mu}_i(t)$. On the other hand, if arm i appears after its correct place, denote the number of times it happens as $T^{a+}_{i,k}$. In that case, at least one arm in the set $\{i+1,i+2,\ldots,K\}$ has larger UCB value than $\hat{\mu}_i(t)$, since at most $K-i$ arms have smaller UCB values than $\hat{\mu}_i(t)$. Thus, $T^{a+}_{i,k}$ can be bounded by the number of times that the maximum UCB value of arms $\{i+1,i+2,\ldots,K\}$ is larger than $\hat{\mu}_i(t)$. We can get:

$$\begin{aligned} T^{a-}_{i,k} &\leq \sum_{t=1}^{T} \mathbb{1}\{\min_{1\leq j\leq i-1} \hat{\mu}_j(t) \leq \hat{\mu}_i(t)\} \\ &\leq \sum_{j=1}^{i-1}\sum_{t=1}^{T} \mathbb{1}\{\hat{\mu}_j(t) \leq \hat{\mu}_i(t)\}. \end{aligned} \tag{4.37}$$

$$\begin{aligned} T^{a+}_{i,k} &\leq \sum_{t=1}^{T} \mathbb{1}\{\max_{i+1\leq j\leq K} \hat{\mu}_j(t) \geq \hat{\mu}_i(t)\} \\ &\leq \sum_{j=i+1}^{K}\sum_{t=1}^{T} \mathbb{1}\{\hat{\mu}_j(t) \geq \hat{\mu}_i(t)\}. \end{aligned} \tag{4.38}$$

Similar to Lemma 4.5, we can bound both terms $T^{a-}_{i,k}$ and $T^{a+}_{i,k}$, and $T^a_{i,k}$ should be

less than their sum:

$$\begin{aligned}
\mathbb{E}[T_{i,k}^a] &\leq \mathbb{E}[T_{i,k}^{a-}] + \mathbb{E}[T_{i,k}^{a+}] \\
&\leq \mathbb{E}\left[\sum_{j=1}^{i-1}\sum_{t=1}^{T} \mathbb{1}\{\hat{\mu}_j(t) \leq \hat{\mu}_i(t)\} + \sum_{j=i+1}^{K}\sum_{t=1}^{T} \mathbb{1}\{\hat{\mu}_j(t) \geq \hat{\mu}_i(t)\}\right] \\
&\leq \sum_{j=1}^{i-1}\left\{\frac{8\log T}{\Delta_{j,i}^2} + 1 + \frac{\pi^2}{3}\right\} + \sum_{j=i+1}^{K}\left\{\frac{8\log T}{\Delta_{i,j}^2} + 1 + \frac{\pi^2}{3}\right\} \\
&\leq (K-1)\,(\frac{8\log T}{\Delta_{\min}^2} + 1 + \frac{\pi^2}{3}),
\end{aligned} \tag{4.39}$$

which concludes the proof. □

With Lemma 4.6 and (4.35), we can write $R_{i,k}^a$ as:

$$\begin{aligned}
\mathbb{E}[R_{i,k}^a(T)] \leq & R_k^{\max}\,\mathbb{E}[T_{i,k}^a] \\
\leq & (L-k+1)(i-1)(\frac{8\log T}{\Delta_{i-1,i}^2} + 1 + \frac{\pi^2}{3})\mu_{1+(j-1)M} \\
& + (L-k+1)(K-i)(\frac{8\log T}{\Delta_{i,i+1}^2} + 1 + \frac{\pi^2}{3})\mu_{1+(j-1)M} \\
\leq & c_\mu(L-k+1)(i-1)\left[\frac{8\log T}{\Delta_{i-1,i}} + (1+\frac{\pi^2}{3})\Delta_{i-1,i}\right] \\
& + c_\mu(L-k+1)(K-i)\left[\frac{8\log T}{\Delta_{i,i+1}} + (1+\frac{\pi^2}{3})\Delta_{i,i+1}\right].
\end{aligned} \tag{4.40}$$

Define $T_{\max} := \frac{8\log T}{\Delta_{\min}} + (1+\frac{\pi^2}{3})\Delta_{\max}$. Finally, with (4.5), (4.36) and (4.40), we can bound $\mathbb{E}[R(T)]$:

$$\begin{aligned}
\mathbb{E}[R(T)] &\leq \sum_{k=1}^{L}\left\{\sum_{i>kM}^{K}\mathbb{E}[R_{i,k}^s(T)] + \sum_{i=(k-1)M+1}^{kM}\mathbb{E}[R_{i,k}^a(T)]\right\} \\
&\leq \sum_{k=1}^{L}\left\{\sum_{i>kM}^{K} c_\mu kM(L-k+1)T_{\max} + \sum_{i=(k-1)M+1}^{kM} c_\mu(L-k+1)(K-1)T_{\max}\right\} \\
&\leq c_\mu L^2K^2\,T_{\max} + \alpha L^2MK\,T_{\max} \\
&\leq c_\mu K^2(L^2+L)\,T_{\max}.
\end{aligned} \tag{4.41}$$

□

4.6.5 Proof of Theorem 4.4

Proof. In order to prove Theorem 4.4, we first consider the following lemma:
Lemma 4.7.

$$\mathbb{E}[Loss(T)] \leq \mu_{\max}\mathbb{E}[\#\ of\ collisions] + \sum_{k=1}^{L}\sum_{i>km}^{K} R_{i,k}^{s}(T) \tag{4.42}$$

Proof. Here $R_{i,k}^{s}$ is as defined in (4.32). Lemma 4.7 essentially upper-bounds Loss(T) by the maximum regret caused by collisions and the total regret due to observing arms in the wrong steps. Whenever there are collisions at any given round t, the expected loss of reward compared to any offline policy is no larger than the highest regret at t over all users who encounter a collision, *i.e.,* every user gets zero reward in our policy while every user gets the highest reward in expectation in the offline policy. When there's no collision, the loss compared to any greedy policy is caused by observing arms in the wrong steps, *i.e.,* which is at most $\sum_{k=1}^{L}\sum_{i>km}^{K} R_{i,k}^{s}(T)$. □

To further upper-bound $\mathbb{E}[\text{Loss}(T)]$, we proceed in the next lemma to upper-bound $\mathbb{E}[\#$ of collisions$]$ across all players. The basic idea of the proof is to consider the number of collisions in: (1) rounds where each player chooses from the correct list of arms in each observation step and (2) rounds in which there exists at least one player having at least one arm in the wrong step. We respectively call these (1) good phases (i.e., sequential rounds where the first condition is satisfied in each round) and (2) bad rounds. The term $K\binom{2M-1}{M}$ upper-bounds the number of collisions of each step in each good phase, and M upper-bounds that in each bad round. Since the number of non-sequential good phases is no larger than the number of bad rounds plus one, the lemma follows.
Lemma 4.8. *The total expected number of collisions is at most*

$$\left(K\binom{2M-1}{M} + M\right) \times \sum_{k=1}^{L}\sum_{i>kM}^{K} \mathbb{E}[T_{i,k}^{s}] \tag{4.43}$$

Proof. It is easy to verify the total number of collisions over all bad rounds are at most M times the total number of those rounds. Thus, in the following, we only need to consider the good phases. In a good phase, every user has the same (and also correct) set of arms to observe in each step i. We simply check how the M users are

"assigned to" the M arms in each step i. We first consider a given round t where every user encounters a collision in round $t-1$. In this case, each user will uniformly at random select one out of those M arms in round t. We now consider the total number of distinct configurations of arms and users. Since in this lemma, we are calculating the number of collisions rather than the reward or regret of each user, we do not distinguish different users choosing the same arm. Thus, two configurations are distinct iff there exists at least one arm that has a different number of assigned users between these two configurations. This random process is equivalent to assigning M balls into M boxes which has a total of $\binom{2M-1}{M}$ distinct configurations [88]. Now we consider the cases where γ out of M users (let $M > \gamma > 0$ will continue to choose the same arms as in the previous round, since there was no collision in the previous round. Similarly, the number of distinct user-arm configurations is at most $\binom{2M-1-\gamma}{M-\gamma}$, which is smaller than $\binom{2M-1}{M}$. Since each user's decision is only dependent on his decision and outcome in the previous round, this random process of assigning users to arms over time is a Markov chain with at most $\binom{2M-1}{M}$ states. Moreover, it's easy to verify that once the process enters a state where users choose different arms in a given step, it will stay in this state, as long as the good phase hasn't transitioned to a bad round. Therefore, this stochastic process is an Absorbing Markov chain with an absorbing time no larger than $\binom{2M-1}{M}$ rounds [59]. Thus, the total number of collisions of each step within each good phase is at most $M\binom{2M-1}{M}$. However, we have to consider an extreme case where for any given observation step i, it enters an absorbing state with a number of $\binom{2M-1}{M}$ rounds, but the chosen arms of all users are realized to be unavailable. Thus, all of them have to enter observation step $i+1$ and the process starts over from a possibly transient state. The worst case is that the above extreme case happens over all K/M observation steps. Therefore, the maximum number of collisions in a good phase is at most $K\binom{2M-1}{M}$. Combining the total number of bad rounds with the number of collisions in each good phase and bad round respectively, the lemma follows. □

Note that the multiplicative term in (4.43), $\mathbb{E}[T^s_{i,k}]$, has been given in (4.34). Putting (4.43) and (4.36) into (4.42), we get Theorem 4.4. While this loss bound is logarithmic in the number of rounds T, like the $O(\frac{K^3}{M}L\log(T))$ regret bound given in Theorem 4.3 for the C-MP-OBP policy, it is combinatorial in M instead of being polynomial in $K = LM$. The lack of coordination in the distributed setting

introduces an additional cost from possible collisions. □

4.6.6 Proof of Theorem 4.5

Proof.

Lemma 4.9. *The total expected regret,*

$$\mathbb{E}[R(T)] \leq \mu_{\max}\mathbb{E}[\#\ \textit{of collisions}] + \sum_{k=1}^{L}\left(\sum_{i>km}^{K} R_{i,k}^{s}(T) + \sum_{i=1(k-1)M+1}^{KM} R_{i,k}^{a}(T)\right)$$

Proof. The total expected regret can be upper-bounded by the sum of the expected loss and the expected regret due to choosing the wrong arm from the right step over all users. Combining the proof in Theorem 4.4 and (4.40), this lemma follows. □

To further upper-bound $R(T)$, we upper-bound the expected number of collisions in the following lemma.

Lemma 4.10. *We have:*

$$\mathbb{E}[\#\ \textit{of collisions}] \leq M\left(\binom{2M-1}{M}+1\right) \times \sum_{k=1}^{L}\left(\sum_{i>kM}^{K}\mathbb{E}[T_{i,k}^{s}] + \sum_{i=(k-1)M+1}^{KM}\mathbb{E}[T_{i,k}^{a}]\right)$$

Proof. Interestingly, the first term in (4.44) (the number of collisions in a good phase) is smaller than that of our fair strategy D-MP-OBP. This can be explained intuitively as follows. According to our D-MP-Adapt-OBP, the decisions of the steps $2, \cdots, L$ are determined by the decisions of step 1 and $f(\cdot)$, given the reward estimations of all arms. Therefore, within a good phase, when the first step becomes collision-free, the following steps will all be collision-free. In this sense, the number of collisions will not increase with the number of observation steps. Consistent with the terminologies used in the proof of Lemma 4.8, we consider each round where there exists a user who either chooses an arm in the wrong observation step (a bad round) or chooses the wrong arm from the right observation step. The analysis for the collisions in the former event is the same as Lemma 4.8. The latter event can be divided into three cases: (1) in the first observation step, multiple users play the same arm; (2) in a later observation step $i > 1$, two or more user choose an unavailable arm j in step $i-1$, and they both choose arm $f(j, \{\hat{\mu_k}\}_{k=1}^{K})$ in step i; (3) in a later observation step

$i > 1$, the user has a different order of arms with at least one other user, *e.g.,* user 1 and user 2 are supposed to choose the arms in the second position and the third position respectively but they both choose arm 2 as user 2 mistakenly ranks arm 2 in the third position. In any one of the above three cases, there is at most one collision encountered by each user in each round. Now we consider the good phases in which users have the same (and correct) order of arms. For the first observation step, there are at most $M\binom{2M-1}{M}$ rounds before entering an absorbing state. Since the positions of arms to choose in each step $i > 1$ are determined by the arms chosen in step 1, observing the arms in each observation step $i > 1$ (only when the arms chosen in the previous observation arm sets are unavailable) does not transition the state from an absorbing state to a transient state. Thus, the total expected number of collisions in a good phase over all steps is still $M\binom{2M-1}{M}$, which does not increase with the number of observation steps. Putting the above together, the lemma follows. □

Combining Lemma 4.9, Lemma 4.10, (4.36), and (4.40), the theorem directly follows. □

4.6.7 Proof of Theorem 4.6

Proof.

Lemma 4.11. *Let $\mathcal{N}_t^{\mathrm{s}}$ be the event that at the beginning of round t, for every arm $(k,a) \in S$, $|\hat{\mu}_{k,a,t-1} - \mu_{k,a}| < \rho_{k,a,t}$. Then for each round $t \geq 1$, $\Pr\{\neg\mathcal{N}_t^{\mathrm{s}}\} \leq 2|S|t^{-2}$.*

Proof. For each round $t \geq 1$, we have

$$\begin{aligned}
\Pr\{\neg\mathcal{N}_t^{\mathrm{s}}\} &= \Pr\left\{\exists(k,a) \in S, |\hat{\mu}_{k,a,t-1} - \mu_{k,a}| \geq \sqrt{\frac{3\ln t}{2T_{k,a,t-1}}}\right\} \\
&\leq \sum_{(k,a)\in S} \Pr\left\{|\hat{\mu}_{k,a,t-1} - \mu_{k,a}| \geq \sqrt{\frac{3\ln t}{2T_{k,a,t-1}}}\right\}. \\
&= \sum_{(k,a)\in S} \sum_{s=1}^{t-1} \Pr\left\{T_{k,a,t-1} = s, |\hat{\mu}_{k,a,t-1} - \mu_{k,a}| \geq \sqrt{\frac{3\ln t}{2T_{k,a,t-1}}}\right\}. \qquad (4.44)
\end{aligned}$$

When $T_{k,a,t-1} = s$, $\hat{\mu}_{k,a,t-1}$ is the average of s i.i.d. random outcomes of arm (k,a).

With Hoeffding's Inequality, we have

$$\Pr\left\{T_{k,a,t-1}=s, |\hat{\mu}_{k,a,t-1}-\mu_{k,a}| \geq \sqrt{\frac{3\ln t}{2T_{k,a,t-1}}}\right\} \leq 2t^{-3}, \tag{4.45}$$

Combining Eq.(4.44) and (4.45), we have $\Pr\{\neg\mathcal{N}_t^{\mathrm{s}}\} \leq 2|S|t^{-2}$. □

We generally follow the proof of Theorem 4 in [3], with the different definition of the base arm. We first introduce a positive real number $M_{k,a}$ for each arm (k,a). Let $\mathcal{F}_t$ be the event $\{r'(\boldsymbol{a}_t, \bar{\boldsymbol{\mu}}) < \alpha \cdot \mathrm{opt}(\bar{\boldsymbol{\mu}})\}$, which represents the oracle fails in round t. Define $M_{\boldsymbol{a}} = \max_{(k,a)\in\boldsymbol{a}} M_{k,a}$ for each action $\boldsymbol{a}$. Define

$$\kappa_T(M,s) = \begin{cases} 2B, & \text{if } s=0, \\ 2B\sqrt{\frac{6\ln T}{s}}, & \text{if } 1 \leq s \leq \ell_T(M), \\ 0, & \text{if } s \geq \ell_T(M)+1, \end{cases}$$

where

$$\ell_T(M) = \left\lfloor \frac{24B^2Q^2\ln T}{M^2} \right\rfloor.$$

We then show that if $\{\Delta_{\boldsymbol{a}_t} \geq M_{\boldsymbol{a}_t}\}$, $\neg\mathcal{F}_t$ and $\mathcal{N}_t^{\mathrm{s}}$ hold, we have

$$\Delta_{\boldsymbol{a}_t} \leq \sum_{(k,a)\in\boldsymbol{a}_t} \kappa_T(M_{k,a}, T_{k,a,t-1}). \tag{4.46}$$

The right hand side of the inequality is non-negative, so it holds naturally if $\Delta_{\boldsymbol{a}_t} = 0$. We only need to consider $\Delta_{\boldsymbol{a}_t} > 0$. By $\mathcal{N}_t^{\mathrm{s}}$ and $\neg\mathcal{F}_t$, we have

$$r'(\boldsymbol{a}_t, \bar{\boldsymbol{\mu}}_t) \geq \alpha \cdot \mathrm{opt}(\bar{\boldsymbol{\mu}}_t) \geq \alpha \cdot \mathrm{opt}(\boldsymbol{\mu}) = r'(\boldsymbol{a}_t, \boldsymbol{\mu}) + \Delta_{\boldsymbol{a}_t}$$

Then by Condition 4.2,

$$\Delta_{\boldsymbol{a}_t} \leq r'(\boldsymbol{a}_t, \bar{\boldsymbol{\mu}}_t) - r'(\boldsymbol{a}_t, \boldsymbol{\mu}) \leq B \sum_{(k,a)\in\boldsymbol{a}_t} (\bar{\mu}_{k,a,t} - \mu_{k,a}).$$

We are going to bound $\Delta_{\boldsymbol{a}_t}$ by bounding $\bar{\mu}_{k,a,t} - \mu_{k,a}$. We have

$$\begin{aligned}
\Delta_{\boldsymbol{a}_t} &\leq B \sum_{(k,a)\in\boldsymbol{a}_t} (\bar{\mu}_{k,a,t} - \mu_{k,a}) \\
&\leq -M_{\boldsymbol{a}_t} + 2B \sum_{(k,a)\in\boldsymbol{a}_t} (\bar{\mu}_{k,a,t} - \mu_{k,a}) \\
&\leq 2B \sum_{(k,a)\in\boldsymbol{a}_t} \left[(\bar{\mu}_{k,a,t} - \mu_{k,a}) - \frac{M_{\boldsymbol{a}_t}}{2BQ} \right] \\
&\leq 2B \sum_{(k,a)\in\boldsymbol{a}_t} \left[(\bar{\mu}_{k,a,t} - \mu_{k,a}) - \frac{M_{k,a}}{2BQ} \right]. \qquad (4.47)
\end{aligned}$$

By $\mathcal{N}_t^{\mathrm{s}}$, we have $\bar{\mu}_{k,a,t} - \mu_{k,a} \leq 2\rho_{k,a,t}$, so

$$(\bar{\mu}_{k,a,t} - \mu_{k,a}) - \frac{M_{k,a}}{2BQ} \leq 2\rho_{k,a,t} - \frac{M_{k,a}}{2BQ} \leq 2\sqrt{\frac{3\ln T}{2T_{k,a,t-1}}} - \frac{M_{k,a}}{2BQ}.$$

If $T_{k,a,t-1} \leq \ell_T(M_{k,a})$, we have $(\bar{\mu}_{k,a,t} - \mu_{k,a}) - \frac{M_{k,a}}{2BQ} \leq 2\sqrt{\frac{3\ln T}{2T_{k,a,t-1}}} \leq \frac{1}{2B}\kappa_T(M_{k,a}, T_{k,a,t-1})$. If $T_{k,a,t-1} \geq \ell_T(M_{k,a}) + 1$, then $2\sqrt{\frac{3\ln T}{2T_{k,a,t-1}}} \leq \frac{M_{k,a}}{2BQ}$, so $(\bar{\mu}_{k,a,t} - \mu_{k,a}) - \frac{M_{k,a}}{2BQ} \leq 0 = \frac{1}{2B}\kappa_T(M_{k,a}, T_{k,a,t-1})$. In conclusion, we have

$$(4.47) \leq \sum_{(k,a)\in\boldsymbol{a}_t} \kappa_T(M_{k,a}, T_{k,a,t-1}).$$

Then for all rounds,

$$\begin{aligned}
\sum_{t=1}^{T} \mathbb{I}(\{\Delta_{\boldsymbol{a}_t} \geq M_{\boldsymbol{a}_t}\} \wedge \neg\mathcal{F}_t \wedge \mathcal{N}_t^{\mathrm{s}}) \cdot \Delta_{\boldsymbol{a}_t} &\leq \sum_{t=1}^{T} \sum_{(k,a)\in\boldsymbol{a}_t} \kappa_T(M_{k,a}, T_{k,a,t-1}) \\
&= \sum_{(k,a)\in S} \sum_{s=0}^{T_{k,a,T}} \kappa_T(M_{k,a}, s) \\
&\leq \sum_{(k,a)\in S} \sum_{s=0}^{\ell_T(M_{k,a})} \kappa_T(M_{k,a}, s) \\
&= 2B|S| + \sum_{(k,a)\in S} \sum_{s=1}^{\ell_T(M_{k,a})} 2B\sqrt{\frac{6\ln T}{s}} \\
&\leq 2B|S| + \sum_{(k,a)\in S} \int_{s=0}^{\ell_T(M_{k,a})} 2B\sqrt{\frac{6\ln T}{s}}\mathrm{d}s \\
&\leq 2B|S| + \sum_{(k,a)\in S} 4B\sqrt{6\ln T \ell_T(M_{k,a})} \\
&\leq 2B|S| + \sum_{(k,a)\in S} \frac{48B^2Q\ln T}{M_{k,a}}.
\end{aligned}$$

So

$$\begin{aligned}
Reg(\{\Delta_{\boldsymbol{a}_t} \geq M_{\boldsymbol{a}_t}\} \wedge \neg\mathcal{F}_t \wedge \mathcal{N}_t^{\mathrm{s}}) &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}(\{\Delta_{\boldsymbol{a}_t} \geq M_{\boldsymbol{a}_t}\} \wedge \neg\mathcal{F}_t \wedge \mathcal{N}_t^{\mathrm{s}}) \cdot \Delta_{\boldsymbol{a}_t}\right] \\
&\leq 2B|S| + \sum_{(k,a)\in S} \frac{48B^2Q\ln T}{M_{k,a}}.
\end{aligned}$$

By Lemma 4.11, $\Pr\{\neg\mathcal{N}_t^{\mathrm{s}}\} \leq 2|S|t^{-2}$. Then we have

$$Reg(\neg\mathcal{N}_t^{\mathrm{s}}) \leq \sum_{t=1}^{T} 2|S|t^{-2} \cdot \Delta_{\max} \leq \frac{\pi^2}{3}|S| \cdot \Delta_{\max},$$

$$Reg(\mathcal{F}_t) \leq (1-\beta)T \cdot \Delta_{\max}.$$

With these two bounds, we have

$$\begin{aligned} Reg(\{\}) &\leq Reg(\mathcal{F}_t) + Reg(\neg\mathcal{N}_t^{\mathrm{s}}) + Reg(\{\Delta_{\boldsymbol{a}_t} \geq M_{\boldsymbol{a}_t}\} \wedge \neg\mathcal{F}_t \wedge \mathcal{N}_t^{\mathrm{s}}) + Reg(\Delta_{\boldsymbol{a}_t} < M_{\boldsymbol{a}_t}) \\ &\leq (1-\beta)T \cdot \Delta_{\max} + \frac{\pi^2}{3}|S| \cdot \Delta_{\max} + 2B|S| + \sum_{(k,a)\in S} \frac{48B^2Q\ln T}{M_{k,a}} + Reg(\Delta_{\boldsymbol{a}_t} < M_{\boldsymbol{a}_t}). \end{aligned}$$

Since $Reg_{\alpha,\beta}(T, \boldsymbol{D}) = Reg(\{\}) - (1-\beta)T \cdot \Delta_{\max}$,

$$Reg_{\alpha,\beta}(T, \boldsymbol{D}) \leq \frac{\pi^2}{3}|S| \cdot \Delta_{\max} + 2B|S| + \sum_{(k,a)\in S} \frac{48B^2Q\ln T}{M_{k,a}} + Reg(\Delta_{\boldsymbol{a}_t} < M_{\boldsymbol{a}_t}).$$

For the distribution-dependent bound, take $M_{k,a} = \Delta_{\min}^{k,a}$, then $Reg(\Delta_{\boldsymbol{a}_t} < M_{\boldsymbol{a}_t}) = 0$ and we have

$$Reg_{\alpha,\beta}(T, \boldsymbol{D}) \leq \sum_{(k,a)\in S} \frac{48B^2Q\ln T}{\Delta_{\min}^{k,a}} + 2BKN + \frac{\pi^2}{3} \cdot KN \cdot \Delta_{\max}.$$

For the distribution-independent bound, take $M_{k,a} = M = \sqrt{(48B^2QKN\ln T)/T}$, then $Reg(\Delta_{\boldsymbol{a}_t} < M_{\boldsymbol{a}_t}) \leq TM$ and we have

$$\begin{aligned} Reg_{\alpha,\beta}(T, \boldsymbol{D}) &\leq \sum_{(k,a)\in S} \frac{48B^2Q\ln T}{M_{k,a}} + 2BKN + \frac{\pi^2}{3} \cdot KN \cdot \Delta_{\max} + Reg(\Delta_{\boldsymbol{a}_t} < M_{\boldsymbol{a}_t}) \\ &\leq \frac{48B^2QKN\ln T}{M} + 2BKN + \frac{\pi^2}{3} \cdot KN \cdot \Delta_{\max} + TM \\ &= 2\sqrt{48B^2QKNT\ln T} + \frac{\pi^2}{3} \cdot KN \cdot \Delta_{\max} + 2BKN \\ &\leq 14B\sqrt{QKNT\ln T} + \frac{\pi^2}{3} \cdot KN \cdot \Delta_{\max} + 2BKN. \end{aligned}$$

□

4.6.8 Proof of Theorem 4.8

Proof.

Claim 4.1. *If* $U_{(i,k)} \geq \mu_{(i,k)} \quad \forall (i,k) \in \mathcal{K} \times \mathcal{A}$ *and the UCB-RA and Corr-UCB-RA*

algorithms select a budget allocation $\boldsymbol{S_t}$ *at round* t *where,*

$$\mu_{(i,k)} \leq U_{(i,k)} < \bar{\mu}_{(i,k)} \quad \forall (i,k) \in \mathcal{S}_t,$$

then $\boldsymbol{S_t}$ *is equal to the oracle's optimal allocation* $\boldsymbol{S}^*$*. Here, the thresholds* $\bar{\mu}_{(i,k)}$ *are defined as*

$$\bar{\mu}_{(i,k)} = \mu_{(i,k)} + g^{-1}(\Delta_{min}^{(i,k)}).$$

Proof of Claim 1. In total there are $|K| \times |A|$ base arms. Index these base arms with indices z in the set $\{1, 2, \ldots |K| \times |A|\}$ such that $\Delta_{\min}^{(1)} \geq \Delta_{\min}^{(2)} \geq \ldots \Delta_{\min}^{(z)} \geq \ldots \geq \Delta_{\min}^{(|K|\times|A|)}$.

We consider a case where $\mu_z \leq U_z(t) < \mu_z + g^{-1}(\Delta_{\min}^{(z)}) \quad \forall z \in \mathcal{S}_t$ and $U_z > \mu_z \forall z$. Define y to be the smallest index such that base arm y is selected in $\boldsymbol{S}_t$. From the definition of base arm y and through Condition 4.2 we have,

$$||U_{\boldsymbol{S_t}}(t) - \mu_{\boldsymbol{S_t}}||_\infty < g^{-1}(\Delta_{\min}^{(y)}) \tag{4.48}$$

$$\Rightarrow |r(\boldsymbol{S_t}, \boldsymbol{U(t)}) - r(\boldsymbol{S_t}, \boldsymbol{\mu})| < \Delta_{\min}^{(y)}. \tag{4.49}$$

As $U_z(t) > \mu_z \ \forall z$, we have the following from the monotonicity condition,

$$r(\boldsymbol{S_t}, \boldsymbol{\mu}) + \Delta_{\min}^{(y)} > r(\boldsymbol{S_t}, \boldsymbol{U(t)}) \tag{4.50}$$

$$\geq r(\boldsymbol{S^*}, \boldsymbol{U(t)}) \tag{4.51}$$

$$\geq r(\boldsymbol{S^*}, \boldsymbol{\mu}) \tag{4.52}$$

Here, we have (4.51) as the allocation $\boldsymbol{S_t}$ is obtained from offline oracle and hence it is optimal for the UCB index vector, and its expected reward is larger than the allocation $\boldsymbol{S^*}$. (4.52) arises from the monotonicity condition as $U_z > \mu_z \forall z$. This shows that if $\mu_z \leq U_z(t) < \mu_z + g^{-1}(\Delta_{\min}^{(1)}) \quad \forall z \in S_t$ and $U_z > \mu_z \ \forall z$, then the expected reward for the budget allocation S_t,

$$r(\boldsymbol{S_t}, \boldsymbol{\mu}) > r(\boldsymbol{S^*}, \boldsymbol{\mu}) - \Delta_{\min}^{(y)}. \tag{4.53}$$

As base arm y is selected in $\boldsymbol{S_t}$, then by definition of $\Delta_{min}^{(y)}$,

$$\max(r(\boldsymbol{S_t}, \boldsymbol{\mu})|\boldsymbol{S_t} \in \mathcal{S}_B, (i,k) = y \in \mathcal{S}_t) \leq r(\boldsymbol{S^*}, \boldsymbol{\mu}) - \Delta_{\min}^{(y)}, \tag{4.54}$$

which shows that the maximum reward that can be attained if the allocation $\boldsymbol{S_t}$ was sub-optimal and base arm y was selected is upper bounded by $r(\boldsymbol{S^*}, \boldsymbol{\mu}) - \Delta_{\min}^{(y)}$. Upon comparing (4.54) and (4.53), we conclude that if $\mu_z \leq U_z(t) < \mu_z + g^{-1}(\Delta_{\min}^{(z)}) \quad \forall z \in \mathcal{S}_t$ and $U_z > \mu_z \forall z$, then the budget allocation vector $\boldsymbol{S}_t$ is equal to $\boldsymbol{S}^*$, which is the oracle's unique optimal solution to the budget allocation problem.

Proof of Theorem 1. We now discuss the regret analysis of Theorem 4.8. In order to bound the regret, we first define the notion of a *responsible* base arm.

Definition 4.5 (Responsible)**.** *A base arm (i,k) is responsible at round t, if*

1. *It was selected in round t and*
2. $U_{(i,k)}(t) \geq \bar{\mu}_{(i,k)}$

By Claim 1, if a sub-optimal budget allocation was selected in round t, it implies that either $U_{(i,k)}(t) < \mu_{(i,k)}$ for some $(i,k) \in \mathcal{K} \times \mathcal{A}$ or at least one of the selected base arms in $\boldsymbol{S_t}$ was responsible. Therefore, the expected number of rounds in which a sub-optimal allocation was played (referred to as bad rounds) can be upper bounded by

$$\begin{aligned}\mathbb{E}[\text{Bad rounds}(T)] \leq & \sum_{(i,k)\in\mathcal{K}\times\mathcal{A}} \mathbb{E}[r_{(i,k)}(T)] \\ & + \sum_{(i,k)\in\mathcal{K}\times\mathcal{A}} \mathbb{E}[n_{U_{(i,k)}<\mu_{(i,k)}}(T)],\end{aligned} \tag{4.55}$$

with $r_{(i,k)}(T)$ denoting the number of rounds for which base arm (i,k) is responsible up until round T and $n_{U_{(i,k)}<\mu_{(i,k)}}(T)$ representing the number of rounds in which $U_{(i,k)}(t) < \mu_{(i,k)}$ for some (i,k) until round T. This inequality arises as a result of the union bound and linearity of expectation. Moreover, whenever arm (i,k) is responsible in round t, the regret incurred in that round can be upper bounded by by $\Delta_{\max}^{(i,k)}$ (by definition of $\Delta_{\max}^{(i,k)}$ in Theorem 4.6). In scenarios where $U_{(i,k)}(t) < \mu_{(i,k)}$ for some (i,k), the regret incurred in that round can be upper bounded by $\Delta_{\max}$ (by definition of $\Delta_{\max}$ in Theorem 4.6). Using this observation, we can now bound the

regret as

$$\mathbb{E}[\text{Reg}(T)] \leq \sum_{(i,k)\in\mathcal{K}\times\mathcal{A}} \mathbb{E}[r_{(i,k)}(T)] \times \Delta_{\max}^{(i,k)} + \sum_{(i,k)\in\mathcal{K}\times\mathcal{A}} \mathbb{E}[n_{U_{(i,k)}<\mu_{(i,k)}}(T)] \times \Delta_{\max}. \tag{4.56}$$

Using Hoeffding's inequality, it can be shown that the second term is upper bounded by an O(1) constant, the details are presented in Lemma 7 in the appendix. To bound the regret in (4.56), we bound $\mathbb{E}r_{(i,k)}(T)$ separately for non-competitive and competitive base arms. More specifically, we show that $\mathbb{E}r_{(i,k)}(T)$ is upper bounded by an O(1) constant for non-competitive base arms and is O($\log T$) for competitive base arms. There are two key components to show upper bounds on $\mathbb{E}r_{(i,k)}(T)$ for non-competitive base arm (i,k). Suppose the base arm is non-competitive with respect to (j,ℓ), i.e., $\phi_{(i,k),(j,\ell)} < \bar{\mu}_{(i,k)}$ and $(j,\ell) \in \mathcal{S}^*$.

1. The probability of base arm (i,k) being responsible in round t jointly with the event that $n_{j,\ell}(t) > \frac{2t}{3}$ is *small*.

$$\Pr\left((resp_{(i,k)}(t), n_{(j,\ell)}(t) \geq \frac{2t}{3}\right) \leq t^{-3} \quad \forall t > 3KAt_0.$$

   This occurs as upon obtaining a *large* number of samples of base arm (j,ℓ), the expected pseudo-reward of base arm (i,k) is smaller than $\bar{\mu}_{(i,k)}$ with high probability. As a result, the probability that base arm (i,k) is responsible is *small*. The details of this can be seen in Lemma 4.

2. The probability that a sub-optimal budget allocation is made for more than $\frac{t}{3}$ times till round t is upper bounded as,

$$\Pr\left(T^{\text{sub-opt}}(t) \geq \frac{t}{3}\right) \leq 6(KA)^2 \left(\frac{t}{3KA}\right)^{-2} \quad \forall t > 3KAt_0,$$

   We show this in Lemma 9 through Lemma 6,8 by showing that $r_{(i,k)}(T)$, which is the number of rounds for which base arm (i,k) is responsible till round T, is smaller than $\frac{t}{3KA}$ with high probability. Additionally, $n_{U_{(i,k)}<\mu_{(i,k)}}(T)$, representing the number of rounds in which $U_{(i,k)}(t) < \mu_{(i,k)}$ for some (i,k) till

round T, is smaller than $\frac{t}{3}$ with high probability. Using these two arguments (1) and (2) above, we bound the expected times a non-competitive base arm (i,k) is responsible until round t in Lemma 10 as

$$\mathbb{E}r_{(i,k)}(T) \leq 3KAt_0 + \sum_{t=3KAt_0}^{T} t^{-3} + 6(KA)^2 \left(\frac{t}{3KA}\right)^{-2} \tag{4.57}$$

$$= \mathrm{O}(1). \tag{4.58}$$

Next, we bound the term $\mathbb{E}r_{(i,k)}(T)$ for competitive sub-optimal arms. We do so in Lemma 11, by showing that after base arm (i,k) has been sampled $\mathrm{O}(\log T)$ times, the probability of base arm being responsible at round t decays as t^{-2} and as a result $\mathbb{E}r_{(i,k)}(T)$ is $\mathrm{O}(\log T)$. This combined with (4.58), leads to Theorem 4.8.

□

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we study the competitive CMAB problem from two different perspectives. We first review the traditional non-competitive CMAB problem and propose an algorithm with improved regret bounds. We then introduce competitive CMAB from the follower's perspective, where a follower and a competitor play with the same set of arms. We formulate it as a general C^2MAB-T problem and propose bandit algorithms with tradeoffs between prior knowledge, feedback level, computation efficiency, and regret bound. We have an in-depth study on its application to the online competitive influence maximization (OCIM) problem. We also introduce competitive CMAB from the multi-players' perspective, where multiple players choose combinatorial actions on the same set of arms. We formulate the centralized and distributed settings and study both in the dynamic channel allocation problem. We propose bandit algorithms for different settings and prove their theoretical regret bounds. We also study the general online resource allocation problem, which can be viewed as a centralized problem. We design CMAB algorithms to solve this problem with discrete or continuous budget allocations and discuss how to improve the regret bounds with correlated information.

5.2 Future Work

As discussed in Chapter 2, whether the regret bound improvements in non-competitive CMAB can be extended to the competitive settings is still an open problem. For competitive CMAB from the follower's perspective, though we discuss probabilistic competitor's seed selection in the OCIM problem, it is not clear how to handle the unknown competitor's action in the general setting. We believe the key step is to find a proper (modified) TPM condition and use it to control the incurred regret. For competitive CMAB from the multi-players' perspective, it is interesting to consider players' arrivals and departures. For the online resource allocation problem, we only consider random variables in the objective function; in real-world applications, the random variables may also appear in the constraints. Another interesting direction is to study correlated information in general CMAB problems: although CMAB algorithms learn from the base arm level to avoid exponential explorations on super arms, the number of base arms can still be huge, making it impractical to learn about these base arms within limited rounds. With correlated information across base arms, it might be possible to further reduce the regret bound.

Bibliography

[1] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012. Cited on pages 1 and 90.

[2] Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *The Journal of Machine Learning Research*, 17(1):1746–1778, 2016. Cited on pages 1, 5, 10, 12, 13, 43, 44, 45, 48, 53, 105, 106, and 109.

[3] Qinshi Wang and Wei Chen. Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications. In *Advances in Neural Information Processing Systems*, pages 1161–1171, 2017. Cited on pages 1, 5, 10, 11, 12, 13, 14, 16, 18, 20, 21, 26, 37, 38, 39, 44, 45, 46, 48, 49, 53, 63, 70, 72, 73, 74, 75, 106, 109, 110, and 136.

[4] Zheng Wen, Branislav Kveton, Michal Valko, and Sharan Vaswani. Online influence maximization under independent cascade model with semi-bandit feedback. In *Advances in neural information processing systems*, pages 3022–3032, 2017. Cited on pages 1, 5, 43, and 44.

[5] Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. Cascading bandits: Learning to rank in the cascade model. In *International Conference on Machine Learning*, pages 767–776, 2015. Cited on pages 1, 3, 88, and 90.

[6] Etienne Boursier and Vianney Perchet. Sic-mmab: synchronisation involves communication in multiplayer multi-armed bandits. *Advances in Neural Information Processing Systems*, 32, 2019. Cited on pages 1 and 5.

[7] Pierre Perrault, Jennifer Healey, Zheng Wen, and Michal Valko. Budgeted online influence maximization. In *International Conference on Machine Learning*, pages

7620–7631. PMLR, 2020. Cited on pages 1, 18, 20, 44, and 45.

[8] Chengshuai Shi, Wei Xiong, Cong Shen, and Jing Yang. Decentralized multi-player multi-armed bandits with no collision information. In *International Conference on Artificial Intelligence and Statistics*, pages 1519–1528. PMLR, 2020. Cited on page 5.

[9] Jinhang Zuo, Xutong Liu, Carlee Joe-Wong, John CS Lui, and Wei Chen. Online competitive influence maximization. In *International Conference on Artificial Intelligence and Statistics*, pages 11472–11502. PMLR, 2022. Cited on page 5.

[10] Jinhang Zuo, Xiaoxi Zhang, and Carlee Joe-Wong. Observe before play: Multi-armed bandit with pre-observations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7023–7030, 2020. Cited on page 6.

[11] Shuai Li, Baoxiang Wang, Shengyu Zhang, and Wei Chen. Contextual combinatorial cascading bandits. In *International conference on machine learning*, pages 1245–1253. PMLR, 2016. Cited on pages 6 and 90.

[12] Jinhang Zuo and Carlee Joe-Wong. Combinatorial multi-armed bandits for resource allocation. In *The 55th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–4. IEEE, 2021. Cited on pages 7 and 113.

[13] Samarth Gupta, Jinhang Zuo, Carlee Joe-Wong, Gauri Joshi, and Osman Yağan. Correlated combinatorial bandits for online resource allocation. *The 23rd International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc)*, 2022. Cited on pages 7 and 105.

[14] Xutong Liu, Jinhang Zuo, Siwei Wang, Carlee Joe-Wong, John Lui, and Wei Chen. Batch-size independent regret bounds for combinatorial semi-bandits with probabilistically triggered arms or independent arms. *arXiv preprint arXiv:2208.14837*, 2022. Cited on pages 10 and 14.

[15] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, pages 151–159, 2013. Cited on pages 11 and 12.

[16] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *AISTATS*, 2015. Cited on page 12.

[17] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvári. Combinatorial cascading bandits. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 1450–1458, 2015. Cited on pages 12 and 90.

[18] Nadav Merlis and Shie Mannor. Batch-size independent regret bounds for the combinatorial multi-armed bandit problem. In *Conference on Learning Theory*, pages 2465–2489. PMLR, 2019. Cited on pages 13, 14, 16, 17, 18, 20, and 22.

[19] Nadav Merlis and Shie Mannor. Tight lower bounds for combinatorial multi-armed bandits. *Proceedings of Thirty Third Conference on Learning Theory*, 2020. Cited on pages 18 and 53.

[20] Siwei Wang and Wei Chen. Thompson sampling for combinatorial semi-bandits. In *International Conference on Machine Learning*, pages 5114–5122. PMLR, 2018. Cited on pages 18 and 45.

[21] Rémy Degenne and Vianney Perchet. Combinatorial semi-bandit with known covariance. In *Advances in Neural Information Processing Systems*, pages 2972–2980, 2016. Cited on page 20.

[22] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009. Cited on page 20.

[23] Devdatt P Dubhashi and Alessandro Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2009. Cited on page 20.

[24] Lixing Chen, Jie Xu, and Zhuo Lu. Contextual combinatorial multi-armed bandits with volatile arms and submodular reward. *Advances in Neural Information Processing Systems*, 31:3247–3256, 2018. Cited on pages 36 and 44.

[25] Lijing Qin, Shouyuan Chen, and Xiaoyan Zhu. Contextual combinatorial bandit and its application on diversified online recommendation. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 461–469. SIAM, 2014. Cited on pages 36 and 44.

[26] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146,

2003. Cited on pages 42, 44, 47, and 56.

[27] Tim Carnes, Chandrashekhar Nagarajan, Stefan M Wild, and Anke Van Zuylen. Maximizing influence in a competitive social network: a follower's perspective. In *Proceedings of the ninth international conference on Electronic commerce*, pages 351–360, 2007. Cited on pages 42 and 44.

[28] Shishir Bharathi, David Kempe, and Mahyar Salek. Competitive influence maximization in social networks. In *International workshop on web and internet economics*, pages 306–311, 2007. Cited on pages 42 and 44.

[29] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th international conference on World wide web*, pages 665–674, 2011. Cited on pages 42, 44, 46, and 47.

[30] Xinran He, Guojie Song, Wei Chen, and Qingye Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. In *Proceedings of the 2012 siam international conference on data mining*, pages 463–474, 2012. Cited on pages 42 and 44.

[31] Sergei Ivanov, Konstantinos Theocharidis, Manolis Terrovitis, and Panagiotis Karras. Content recommendation for viral social influence. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 565–574, 2017. Cited on pages 42 and 44.

[32] Wei Chen, Laks V. S. Lakshmanan, and Carlos Castillo. *Information and Influence Propagation in Social Networks*. Morgan & Claypool Publishers, 2013. Cited on pages 42 and 43.

[33] Yuchen Li, Ju Fan, Yanhao Wang, and Kian-Lee Tan. Influence maximization on social graphs: A survey. *IEEE Trans. Knowl. Data Eng.*, 30(10):1852–1872, 2018. Cited on page 44.

[34] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. Maximizing social influence in nearly optimal time. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 946–957, 2014. Cited on page 44.

[35] Youze Tang, Yanchen Shi, and Xiaokui Xiao. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD*

International Conference on Management of Data, pages 1539–1554, 2015. Cited on page 44.

[36] Hung T. Nguyen, My T. Thai, and Thang N. Dinh. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *SIGMOD*, pages 695–710, 2016. Cited on page 44.

[37] Jing Tang, Xueyan Tang, Xiaokui Xiao, and Junsong Yuan. Online processing algorithms for influence maximization. In *SIGMOD*, pages 991–1005, 2018. Cited on page 44.

[38] Yishi Lin and John CS Lui. Analyzing competitive influence maximization problems with partial information: An approximation algorithmic framework. *Performance Evaluation*, 91:187–204, 2015. Cited on pages 44, 51, 53, and 55.

[39] Qingyun Wu, Zhige Li, Huazheng Wang, Wei Chen, and Hongning Wang. Factorization bandits for online influence maximization. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 636–646, 2019. Cited on page 44.

[40] Sharan Vaswani, Branislav Kveton, Zheng Wen, Mohammad Ghavamzadeh, Laks VS Lakshmanan, and Mark Schmidt. Model-independent online learning for influence maximization. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3530–3539, 2017. Cited on page 44.

[41] Alihan Hüyük and Cem Tekin. Thompson sampling for combinatorial network optimization in unknown environments. *IEEE/ACM Transactions on Networking*, 28(6):2836–2849, 2020. Cited on page 45.

[42] Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of thompson sampling. *The Journal of Machine Learning Research*, 17(1):2442–2471, 2016. Cited on page 45.

[43] Wei Chen, Alex Collins, Rachel Cummings, Te Ke, Zhenming Liu, David Rincon, Xiaorui Sun, Yajun Wang, Wei Wei, and Yifei Yuan. Influence maximization in social networks when negative opinions may emerge and propagate. In *Proceedings of the 2011 siam international conference on data mining*, pages 379–390. SIAM, 2011. Cited on pages 46 and 66.

[44] Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014. Cited

on pages 47 and 51.

[45] Shuai Li, Fang Kong, Kejie Tang, Qizhi Li, and Wei Chen. Online influence maximization under linear threshold model. In *Advances in Neural Information Processing Systems*, 2020. Cited on page 49.

[46] Aurélien Garivier, Tor Lattimore, and Emilie Kaufmann. On explore-then-commit strategies. In *Advances in Neural Information Processing Systems*, pages 784–792, 2016. Cited on page 53.

[47] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 807–816, 2009. Cited on page 56.

[48] Xutong Liu, Jinhang Zuo, Xiaowei Chen, Wei Chen, and John CS Lui. Multi-layered network exploration via random walks: From offline optimization to online learning. In *International Conference on Machine Learning*, pages 7057–7066. PMLR, 2021. Cited on page 57.

[49] Donggyu Yun, Alexandre Proutiere, Sumyeong Ahn, Jinwoo Shin, and Yung Yi. Multi-armed bandit with additional observations. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(1):13, 2018. Cited on page 88.

[50] Jonathan Rosenski, Ohad Shamir, and Liran Szlak. Multi-player bandits–a musical chairs approach. In *International Conference on Machine Learning*, pages 155–163, 2016. Cited on pages 89 and 90.

[51] Lilian Besson and Emilie Kaufmann. Multi-player bandits revisited. In *Algorithmic Learning Theory*, pages 56–92, 2018. Cited on pages 89, 90, and 98.

[52] Rohit Kumar, A Yadav, Sumit Jagdish Darak, and Manjesh K Hanawal. Trekking based distributed algorithm for opportunistic spectrum access in infrastructure-less network. In *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 1–8. IEEE, 2018. Cited on pages 89 and 90.

[53] Bowen Li, Panlong Yang, Jinlong Wang, Qihui Wu, Shaojie Tang, Xiang-Yang Li, and Yunhao Liu. Almost optimal dynamically-ordered channel sensing and accessing for cognitive networks. *IEEE Transactions on Mobile Computing*,

13(10):2215–2228, 2014. Cited on pages 89, 90, and 93.

[54] T.L Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.*, 6(1):4–22, 1985. Cited on page 90.

[55] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256, 2002. Cited on pages 90, 92, 126, and 129.

[56] Sahand Haji Ali Ahmad, Mingyan Liu, Tara Javidi, Qing Zhao, and Bhaskar Krishnamachari. Optimality of myopic sensing in multichannel opportunistic access. *IEEE Transactions on Information Theory*, 55(9):4040–4050, 2009. Cited on page 90.

[57] Lifeng Lai, Hesham El Gamal, Hai Jiang, and H Vincent Poor. Cognitive medium access: Exploration, exploitation, and competition. *IEEE transactions on mobile computing*, 10(2):239–253, 2011. Cited on page 90.

[58] Keqin Liu and Qing Zhao. Distributed learning in multi-armed bandit with multiple players. *IEEE Transactions on Signal Processing*, 58(11):5667–5681, 2010. Cited on page 90.

[59] Animashree Anandkumar, Nithin Michael, Ao Kevin Tang, and Ananthram Swami. Distributed algorithms for learning and cognitive medium access with logarithmic regret. *IEEE Journal on Selected Areas in Communications*, 29(4):731 – 745, 2011. Cited on pages 90, 91, and 133.

[60] Orly Avner and Shie Mannor. Multi-user lax communications: a multi-armed bandit approach. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016. Cited on page 90.

[61] Rémi Bonnefoi, Lilian Besson, Christophe Moy, Emilie Kaufmann, and Jacques Palicot. Multi-armed bandit learning in iot networks: Learning helps even in non-stationary settings. In *International Conference on Cognitive Radio Oriented Wireless Networks*, pages 173–185. Springer, 2017. Cited on page 90.

[62] Orly Avner and Shie Mannor. Concurrent bandits and cognitive radio networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 66–81. Springer, 2014. Cited on page 90.

[63] Richard Combes, Stefan Magureanu, Alexandre Proutiere, and Cyrille Laroche. Learning to rank: Regret lower bounds and efficient algorithms. *ACM SIGMETRICS Performance Evaluation Review*, 43(1):231–244, 2015. Cited on pages 90 and 93.

[64] Shi Zong, Hao Ni, Kenny Sung, Nan Rosemary Ke, Zheng Wen, and Branislav Kveton. Cascading bandits for large-scale recommendation problems. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 835–844. AUAI Press, 2016. Cited on page 90.

[65] Ruida Zhou, Chao Gan, Jing Yan, and Cong Shen. Cost-aware cascading bandits. In *International Joint Conference on Artificial Intelligence*, 2018. Cited on page 90.

[66] Ravindra K Ahuja, Arvind Kumar, Krishna C Jha, and James B Orlin. Exact and heuristic algorithms for the weapon-target assignment problem. *Operations research*, 55(6):1136–1146, 2007. Cited on page 95.

[67] Marzio De Biasi. Weapon-target assignment problem. http://www.nearly42.org/cstheory/weapon-target-assignment-problem/, 2013. Cited on page 95.

[68] Shangxing Wang. Multichannel dqn channel model. https://github.com/ANRGUSC/MultichannelDQN-channelModel, 2018. Cited on page 102.

[69] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Bandits and experts in metric spaces. *Journal of the ACM (JACM)*, 66(4):1–77, 2019. Cited on pages 105, 106, 111, and 112.

[70] Tarek Hegazy. Optimization of resource allocation and leveling using genetic algorithms. *Journal of construction engineering and management*, 125(3):167–175, 1999. Cited on page 106.

[71] David Julian, Mung Chiang, Daniel O'Neill, and Stephen Boyd. Qos and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks. In *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 477–486. IEEE, 2002. Cited on page 106.

[72] Carlee Joe-Wong, Soumya Sen, Tian Lan, and Mung Chiang. Multiresource allocation: Fairness–efficiency tradeoffs in a unifying framework. *IEEE/ACM*

Transactions on Networking, 21(6):1785–1798, 2013. Cited on page 106.

[73] Nikhil R Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher A Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. *Journal of the ACM (JACM)*, 66(1):1–41, 2019. Cited on page 106.

[74] Nikhil R Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher A Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 29–38, 2011. Cited on page 106.

[75] Tor Lattimore, Koby Crammer, and Csaba Szepesvári. Linear multi-resource allocation with semi-bandit feedback. In *NIPS*, pages 964–972, 2015. Cited on page 106.

[76] Arun Verma, Manjesh K Hanawal, Arun Rajkumar, and Raman Sankaran. Censored semi-bandits: A framework for resource allocation with censored feedback. In *NeurIPS*, 2019. Cited on page 106.

[77] Xavier Fontaine, Shie Mannor, and Vianney Perchet. An adaptive stochastic optimization algorithm for resource allocation. In *Algorithmic Learning Theory*, pages 319–363. PMLR, 2020. Cited on page 106.

[78] Samarth Gupta, Gauri Joshi, and Osman Yağan. Correlated multi-armed bandits with a latent random source. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3572–3576. IEEE, 2020. Cited on page 106.

[79] Samarth Gupta, Shreyas Chaudhari, Gauri Joshi, and Osman Yağan. Multi-armed bandits with correlated arms. *IEEE Transactions on Information Theory*, 67(10):6711–6732, 2021. Cited on page 106.

[80] Tor Lattimore and Rémi Munos. Bounded regret for finite-armed structured bandits. In *Advances in Neural Information Processing Systems*, pages 550–558, 2014. Cited on page 106.

[81] Samarth Gupta, Shreyas Chaudhari, Subhojyoti Mukherjee, Gauri Joshi, and Osman Yağan. A unified approach to translate classical bandit algorithms to the structured bandit setting, 2018. Cited on page 106.

[82] Richard Combes, Stefan Magureanu, and Alexandre Proutiere. Minimal exploration in structured stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 1763–1771, 2017. Cited on page 106.

[83] Andi Nika, Sepehr Elahi, and Cem Tekin. Contextual combinatorial volatile multi-armed bandit with adaptive discretization. In *International Conference on Artificial Intelligence and Statistics*, pages 1486–1496, 2020. Cited on page 112.

[84] Yi Gai and Bhaskar Krishnamachari. Online learning algorithms for stochastic water-filling. In *2012 Information Theory and Applications Workshop*, pages 352–356, 2012. Cited on page 119.

[85] Adarsh Narasimhamurthy, Mahesh Banavar, and Cihan Tepedelenliouglu. *OFDM Systems for Wireless Communications*. Morgan & Claypool, 2010. Cited on page 119.

[86] Norman Abramson. The aloha system: Another alternative for computer communications. In *Proceedings of the November 17-19, 1970, fall joint computer conference*, pages 281–285, 1970. Cited on page 121.

[87] Ljubica Pajevic, Gunnar Karlsson, and Viktoria Fodor. CRAWDAD dataset kth/campus (v. 2019-07-01). Downloaded from https://crawdad.org/kth/campus/20190701/eduroam, July 2019. traceset: eduroam. Cited on page 121.

[88] Miklos Bona. *A Walk Through Combinatorics*. World Scientific, 2011. Cited on page 133.