Observe Before Play: Multi-Armed Bandit with Pre-Observations

Jinhang Zuo, Xiaoxi Zhang, Carlee Joe-Wong
Carnegie Mellon University
{jzuo, xiaoxiz2, cjoewong}@andrew.cmu.edu

Abstract

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. Apart from the usual trade-off between exploring new arms to find the best one and exploiting the arm believed to offer the highest reward, 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 Observe-Before-Play Upper Confidence Bound (OBP-UCB) algorithm for K arms with Bernoulli rewards, 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 data and wireless channel traces show that C-MP-OBP and D-MP-OBP outperform random heuristics and offline optimal policies that do not allow pre-observations.

1 Introduction

Multi-armed bandit (MAB) problems have attracted much attention as a means of capturing the trade-off between exploration and exploitation (Bubeck and Cesa-Bianchi 2012) in sequential decision making. 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.

1.1 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. (Yun et al. 2018), 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 (Kveton et al. 2015a), 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. In this paper, 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.

1.2 Applications

Although many MAB works take cognitive radios as their primary motivation (Rosenski, Shamir, and Szlak 2016; Besson and Kaufmann 2018; Kumar et al. 2018), 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.

1.3 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 (Li et al. 2014), and more easily generalizes to multi-player settings. In the rest of the paper, "user" and "player" are interchangeable.

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.

We discuss related work in Section 2 and consider the single-player setting in Section 3. We generalize these results to multiple players in centralized (Section 4) and distributed (Section 5) settings. We numerically validate our results in Section 6 and conclude in Section 7. Due to the space constraint, detailed proofs are moved to the full technical report (Zuo, Zhang, and Joe-Wong 2019).

2 Related Work

Multi-armed Bandit (MAB) problems have been studied since the 1950s (Lai and Robbins 1985; Bubeck and Cesa-Bianchi 2012). (Auer, Cesa-Bianchi, and Fischer 2002), for instance, propose a simple UCB1 policy that achieves logarithmic regret over time. Recently, MAB applications to Cognitive Radio Networks (CRNs) have attracted attention (Ahmad et al. 2009; Lai et al. 2011), especially in multi-player settings (Liu and Zhao 2010; Anandkumar et al. 2011; Avner and Mannor 2016; Bonnefoi et al. 2017; Kumar et al. 2018) where users choose from the same arms (wireless channels). None of these works include pre-observations, though some (Avner and Mannor 2014; Rosenski, Shamir, and Szlak 2016; Besson and Kaufmann 2018) consider distributed settings. (Li et al. 2014; Combes et al. 2015) 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 (Kveton et al. 2015a; 2015b; Zong et al. 2016). The idea of pre-observations with costs is similar to the cost-aware cascading bandits proposed in (Zhou et al. 2018) and contextual combinatorial cascading bandits introduced in (Li et al. 2016). However, in (Zhou et al. 2018), 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.

3 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 (Anandkumar et al. 2011) an i.i.d. Bernoulli reward of each arm to capture the occupancy/vacancy of each channel (arm). Let $Y_{k,t} \overset{iid}{\sim} \text{Bern}(\mu_k) \in \{0,1\}$ de-

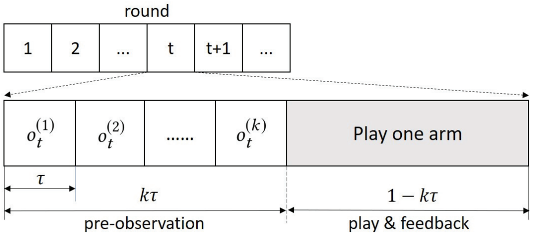


Figure 1: Illustration of Pre-observations

note the reward of arm k at round t, with expected value $\mu_k \in [0,1]$. As shown in Figure 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:

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

$$\mathbb{E}[r(T)] = \sum_{t=1}^{T} \sum_{k=1}^{K} \left\{ (1 - k\,\tau) \mu_{o_t^{(k)}} \prod_{i=1}^{k-1} (1 - \mu_{o_t^{(i)}}) \right\}, \tag{2}$$

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 3.1. *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., $\boldsymbol{o}_t^* = (1, 2, \ldots, K)$.

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 1 and uses UCB values to estimate arm rewards as in traditional MAB algorithms (Auer, Cesa-Bianchi, and Fischer 2002). 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

Algorithm 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
    $\boldsymbol{o}_t$ = 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
```

($Y_{o_t^{(i)},t} = 1$) 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.

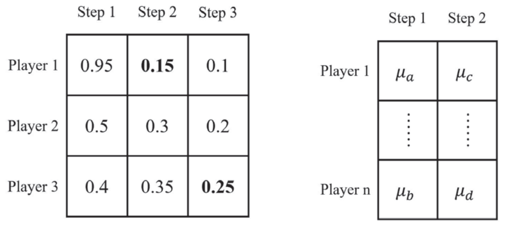
We can define and bound the *regret* of this algorithm as the difference between the expected reward of the optimal policy (Lemma 3.1) and that of the real policy:

$$\begin{aligned} R(T) :=& \mathbb{E}[r^*(T)] - \mathbb{E}[r(T)] \\ =& \sum_{t=1}^{T} \sum_{k=1}^{K} \Bigg\{ (1 - k\,\tau) \mu_k \prod_{i=1}^{k-1} (1 - \mu_i) - \\ & (1 - k\,\tau) \mu_{o_t^{(k)}} \prod_{i=1}^{k-1} (1 - \mu_{o_t^{(i)}}) \Bigg\}. \end{aligned} \tag{3}$$

Theorem 3.2. *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 (Li et al. 2014). However, our proof method is distinct from theirs and preserves the dependence on the arm rewards (through the W_i in Theorem 3.2). 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 (Li et al. 2014) in 99% of our experiments. Moreover, unlike the analysis in (Li et al. 2014), 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 (Combes et al. 2015), but the regret bound of their proposed algorithm has a constant term (independent of T),



	Step 1	Step 2	Step 3
Player 1	0.95	**0.15**	0.1
Player 2	0.5	0.3	0.2
Player 3	0.4	0.35	**0.25**

(a) Non-greedy optimal policy.

	Step 1	Step 2
Player 1	μ_a	μ_c
	⋮	⋮
Player n	μ_b	μ_d

(b) Assigning arms.

Figure 2: Multi-player observation lists, with rewards in the boxes.

$K^2\eta^2$, 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 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 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 $\boldsymbol{o}_{m,t} := (o^{(1)}_{m,t}, o^{(2)}_{m,t}, \ldots, o^{(i)}_{m,t}, \ldots)$ as the **pre-observation list** of player m at round t, where $o^{(i)}_{m,t}$ 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:

$$\mathbb{E}[r(T)] = \sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{k=1}^{|\boldsymbol{o}_{m,t}|}\left\{(1-k\tau)\mu_{o^{(k)}_{m,t}}\prod_{i=1}^{k-1}(1-\mu_{o^{(i)}_{m,t}})\right\}. \quad (4)$$

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 3.1 from the single-player setting:

Lemma 4.1. *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 4.1, 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). 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:

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

where we define $\sum_{\emptyset} := 0$ and $\prod_{\emptyset} := 1$. We show the Weapon Target Assignment (WTA) problem (Ahuja et al. 2007) with identical targets, which is NP-hard (Biasi 2013), 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 4.1 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 $\boldsymbol{s}_{i,t} := (o^{(i)}_{1,t}, o^{(i)}_{2,t}, \ldots, o^{(i)}_{M,t})$ 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,

Algorithm 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
```

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 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 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:

$$R(T) = \sum_{t,m,k=1}^{T,M,L} \Bigg\{ (1-k\tau)\mu_{(k-1)M+m} \prod_{i=1}^{k-1} (1-\mu_{(i-1)M+m}) - (1-k\tau)\mu_{o_{m,t}^{(k)}} \prod_{i=1}^{k-1} (1-\mu_{o_{m,t}^{(i)}}) \Bigg\}. \tag{5}$$

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

Theorem 4.4. *The expected regret of C-MP-OBP is $\mathbb{E}[R(T)] \leq c_\mu K^2(L^2+L)\left(\frac{8\log T}{\Delta_{\min}} + (1+\frac{\pi^2}{3})\Delta_{\max}\right)$, where $\Delta_{\max} = \max\limits_{i<j} \mu_i - \mu_j$, $\Delta_{\min} = \min\limits_{i<j} \mu_i - \mu_j$.*

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 4.4 to match the lower bound of $O(KM\log T)$ given by (Besson and Kaufmann 2018).

5 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; 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.

The D-MP-OBP algorithm is shown in Algorithm 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

Algorithm 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**
    $\boldsymbol{\alpha} =$ 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**
```

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 5.1. *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 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 $o_{m,t}^l$ 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 $o_{m,t}^{l+1} = f(o_{m,t}^l, \{\hat{\mu_k(t)}\}_{k=1}^K), \forall l = 1, ..., L-1$ 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 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 5.2. *The expected regret, $\mathbb{E}[R(T)]$ of our distributed algorithm D-MP-Adapt-OBP is logarithmic in T.*

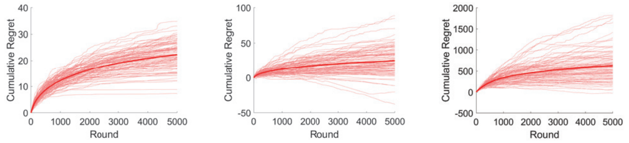
We observe from the proof of Theorem 5.2 that the regret is combinatorial in M but logarithmic in T, unlike the centralized multi-player setting's $O(K^2L^2 \log T)$ regret in Theorem 4.4. This scaling with M comes from the lack of coordination between players and the resulting collisions.

6 Experiments

We validate the theoretical results from Sections 3–5 with numerical simulations. We summarize our results as follows:

Sublinear regret: We show in Figure 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 3.1), 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 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 3c is larger than that of C-MP-OBP, likely due to collisions in the distributed setting.

Superiority to baseline strategies: We show in Tables 1 and 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 (Wang 2018). 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



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

Figure 3: Sublinear regret in each setting. Each line represents an experiment run with randomly chosen reward distributions; the bold line is the average over 100 runs.

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%

Table 1: Average % reward improvements of OBP-UCB

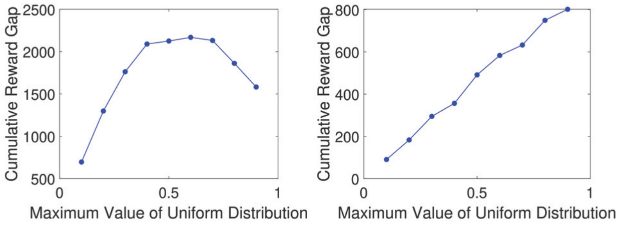
choosing the order of the pre-observations. Our second baseline is an **optimal offline single-observation policy** (**single-opt**), which allocates the arms with the M highest rewards to each player (in the single-player setting, $M = 1$). These optimal offline policies are superior to any learning-based 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 1 and 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 4a and 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 in multi-player settings are shown in the technical report.

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%

Table 2: Average C-MP-OBP, D-MP-OBP % improvement.



(a) Single-observation baseline. **(b)** Random baseline.

Figure 4: Average cumulative reward gaps in the single-player (OBP-UCB) setting after 5000 rounds over 100 experiments, when $\tau = 0.1$ and $K = 9$ arms with expected rewards μ's uniformly drawn from the range $[0, x]$.

7 Discussion and Conclusion

In this work, we introduce **pre-observations** into multi-armed bandit problems. Such pre-observations introduce new technical challenges to the MAB framework, as players must not only learn the best set of arms, but also the optimal order in which to pre-observe these arms. This challenge is particularly difficult in multi-player settings, as each player must learn an observation set of arms that avoids collisions with other players. We develop algorithms for both the single- and multi-player settings and show that they achieve logarithmic regret over multiple rounds. As one of the first works to consider pre-observations, however, we leave several problems open for future work. One might, for instance, consider user arrivals and departures, which would affect the offline optimal observation lists; or temporal reward correlations. Both of these would likely arise in our motivating scenario of cognitive radio networks, as devices move in and out of range and channel incumbents exhibit temporal behavior patterns. Another challenging extension would be to consider cases with more limited collisions, where one arm might serve multiple users (e.g., if an "arm" is a city block when users are searching for parking spaces). In such cases, we must learn not just the probability that the arm is available (i.e., its expected reward) but also the full distribution of the number of users that the arm can accommodate.

References

Ahmad, S. H. A.; Liu, M.; Javidi, T.; Zhao, Q.; and Krishnamachari, B. 2009. Optimality of myopic sensing in multichannel opportunistic access. *IEEE Transactions on Information Theory* 55(9):4040–4050.

Ahuja, R. K.; Kumar, A.; Jha, K. C.; and Orlin, J. B. 2007. Exact and heuristic algorithms for the weapon-target assignment problem. *Operations research* 55(6):1136–1146.

Anandkumar, A.; Michael, N.; Tang, A. K.; and Swami, A. 2011. Distributed algorithms for learning and cognitive medium access with logarithmic regret. *IEEE Journal on Selected Areas in Communications* 29(4):731 – 745.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.* 47(2-3):235–256.

Avner, O., and Mannor, S. 2014. Concurrent bandits and cognitive radio networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 66–81. Springer.

Avner, O., and Mannor, S. 2016. Multi-user lax communications: a multi-armed bandit approach. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, 1–9. IEEE.

Besson, L., and Kaufmann, E. 2018. Multi-player bandits revisited. In *Algorithmic Learning Theory*, 56–92.

Biasi, M. D. 2013. Weapon-target assignment problem. http://www.nearly42.org/cstheory/weapon-target-assignment-problem/.

Bonnefoi, R.; Besson, L.; Moy, C.; Kaufmann, E.; and Palicot, J. 2017. Multi-armed bandit learning in iot networks: Learning helps even in non-stationary settings. In *International Conference on Cognitive Radio Oriented Wireless Networks*, 173–185. Springer.

Bubeck, S., and Cesa-Bianchi, N. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning* 5(1):1–122.

Combes, R.; Magureanu, S.; Proutiere, A.; and Laroche, C. 2015. Learning to rank: Regret lower bounds and efficient algorithms. *ACM SIGMETRICS Performance Evaluation Review* 43(1):231–244.

Kumar, R.; Yadav, A.; Darak, S. J.; and Hanawal, M. K. 2018. 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)*, 1–8. IEEE.

Kveton, B.; Szepesvari, C.; Wen, Z.; and Ashkan, A. 2015a. Cascading bandits: Learning to rank in the cascade model. In *International Conference on Machine Learning*, 767–776.

Kveton, B.; Wen, Z.; Ashkan, A.; and Szepesvari, C. 2015b. Combinatorial cascading bandits. In *Advances in Neural Information Processing Systems*, 1450–1458.

Lai, T., and Robbins, H. 1985. Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.* 6(1):4–22.

Lai, L.; El Gamal, H.; Jiang, H.; and Poor, H. V. 2011. Cognitive medium access: Exploration, exploitation, and competition. *IEEE transactions on mobile computing* 10(2):239–253.

Li, B.; Yang, P.; Wang, J.; Wu, Q.; Tang, S.; Li, X.-Y.; and Liu, Y. 2014. Almost optimal dynamically-ordered channel sensing and accessing for cognitive networks. *IEEE Transactions on Mobile Computing* 13(10):2215–2228.

Li, S.; Wang, B.; Zhang, S.; and Chen, W. 2016. Contextual combinatorial cascading bandits. In *ICML*, volume 16, 1245–1253.

Liu, K., and Zhao, Q. 2010. Distributed learning in multiarmed bandit with multiple players. *IEEE Transactions on Signal Processing* 58(11):5667–5681.

Rosenski, J.; Shamir, O.; and Szlak, L. 2016. Multi-player bandits–a musical chairs approach. In *International Conference on Machine Learning*, 155–163.

Wang, S. 2018. https://github.com/ANRGUSC/MultichannelDQN-channelModel.

Yun, D.; Proutiere, A.; Ahn, S.; Shin, J.; and Yi, Y. 2018. Multi-armed bandit with additional observations. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2(1):13.

Zhou, R.; Gan, C.; Yan, J.; and Shen, C. 2018. Cost-aware cascading bandits. In *International Joint Conference on Artificial Intelligence*.

Zong, S.; Ni, H.; Sung, K.; Ke, N. R.; Wen, Z.; and Kveton, B. 2016. Cascading bandits for large-scale recommendation problems. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, 835–844. AUAI Press.

Zuo, J.; Zhang, X.; and Joe-Wong, C. 2019. Observe before play: Multi-armed bandit with pre-observations. Available at https://research.ece.cmu.edu/lions/Papers/OBP_AAAI.pdf, to appear in arXiv.