The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)

Decentralized Stochastic Multi-Player Multi-Armed Walking Bandits

Guojun Xiong, Jian Li

SUNY-Binghamton University
{gxiong1,lij}@binghamton.edu

Abstract

Multi-player multi-armed bandit is an increasingly relevant decision-making problem, motivated by applications to cognitive radio systems. Most research for this problem focuses exclusively on the settings that players have *full access* to all arms and receive no reward when pulling the same arm. Hence all players solve the same bandit problem with the goal of maximizing their cumulative reward. However, these settings neglect several important factors in many real-world applications, where players have *limited access* to *a dynamic local subset of arms* (i.e., an arm could sometimes be "walking" and not accessible to the player). To this end, this paper proposes a *multi-player multi-armed walking bandits* model, aiming to address aforementioned modeling issues. The goal now is to maximize the reward, however, players can only pull arms from the local subset and only collect a full reward if no other players pull the same arm. We adopt Upper Confidence Bound (UCB) to deal with the exploration-exploitation tradeoff and employ distributed optimization techniques to properly handle collisions. By carefully integrating these two techniques, we propose a decentralized algorithm with near-optimal guarantee on the regret, and can be easily implemented to obtain competitive empirical performance.

Introduction

The multi-armed bandit (MAB) framework has been widely adopted for studying sequential decision-making problems (Robbins 1952; Lai and Robbins 1985; Auer, Cesa-Bianchi, and Fischer 2002; Bubeck and Cesa-Bianchi 2012) in a variety of applications. In a classic MAB setting, the decision maker chooses one arm from the set of $\mathcal{K} = \{1, \cdots, K\}$ arms at each time and receives a random reward according to unknown reward distributions. The rewards of different arms are assumed to be independent and identically distributed over time. The goal of the decision maker is to maximize the cumulative reward in the face of unknown mean rewards.

Recently, there has been an increased interest in studying the MAB in multi-player settings, dubbed as `MPMAB`, where the problem gets more intricate as N independent decision makers (i.e., players) are involved. At each discrete time t, each player selects one arm from $\mathcal{K}$, receives some feedback about this arm and possibly shares some "information" with her neighbors. Two popular settings have been widely studied: a collision setting, where a player collects the full reward from the selected arm only if no other players pull the same arm, as motivated by radio channel assignment in cognitive radios (Jouini et al. 2009); and a collaborative setting, where players receive independent reward when they pull the same arm, and cooperatively solve a MAB, as motivated by sequential decisions in social networks (Landgren, Srivastava, and Leonard 2016). In this work, we focus on the former setting, and simply refer to it as the `MPMAB`.

However, the basic model for `MPMAB` in most prior works assumes that players have *full access* to all K arms in each time. This neglects several important factors of systems for many real-world applications, where each player can *only* access a *subset of arms* that *dynamically* changes over time (i.e., an arm could sometimes be "walking" and not accessible to the player). For example, consider the problem of content placement in next-generation wireless networks (e.g., 5G/6G) (Andrews et al. 2014) where N cache-enabled base stations (players) serve a region where mobile users request for K contents (arms), e.g., movies, videos, etc. Users receive a large reward (e.g., a short latency) if the requested content is stored in the nearest base station, otherwise they are served by farther base stations with a small reward (e.g., a larger latency). The base stations initially have no information about users' content requests and contents' global popularity since each base station only have access to a subset of contents due to its constrained cache size. In reality, users' content requests are highly dynamic and hence each base station needs to repeatedly determine the subset of contents to be cached so as to maximize the total reward of serving users. Another application is mobile edge computing (Ceselli, Premoli, and Secci 2017; Farhadi et al. 2021), where edge clouds (arms) with computing resources form a shared resource pool, which can be allocated among user requests (players) that only have access to some edge clouds within the same geographical region. Additional real-world applications where players only have access to a subset of arms are presented in (Xiong and Li 2022).

In this paper, we introduce a new bandit model in formalizing the *walking arms* such that "each player only accesses a subset of arms that dynamically changes over time". Specifically, at time t, player i only has access to a subset $\mathcal{S}_i(t) \subseteq \mathcal{K}$ of arms, where $\mathcal{S}_i(t)$ is changing over time. As a

result, we call the set of $\mathcal{K}$ arms as "*walking arms*" and refer to the subset $\mathcal{S}_i(t)$ as *the local walking arm set*. The goal is to find the optimal arm in $\mathcal{S}_i(t)$ for each player $i \in \mathcal{N}$ at each time t to maximize the cumulative reward over a finite time horizon T. However, player i only observes a full reward if no other players pull the same arm. We call this new bandit model as *"multi-player multi-armed walking bandits"* (MPMAB-WA).

To the best of our knowledge, this is the first work that integrates all three critical factors of multiple players, collisions and dynamic local walking arms into a unified MPMAB model. However, the MPMAB-WA problem becomes much more challenging. In particular, the dynamic local walking arms introduce an additional layer of complexity to the MPMAB problem that is already quite intricate. This is because each player not only encounters a non-trivial tradeoff between *exploration* (i.e., seeking better options) and *exploitation* (i.e., staying with the currently-known best option) when attempting to maximizing the reward, but also is faced with *a new dilemma* of how to manage the balance between maximizing the reward and avoiding collisions when players only receive feedback from a dynamic local subset of arms at each time.

Though several known MPMAB algorithms can successfully handle the exploration-exploitation tradeoff, this new dilemma make existing arm elimination (Lykouris, Mirrokni, and Paes Leme 2018; Gupta et al. 2021; Boursier and Perchet 2019), learning-to-rank (Combes et al. 2015; Tibrewal et al. 2019) and leader-follower (Wang et al. 2020; Mehrabian et al. 2020) methods inapplicable in MPMAB-WA. In this paper, we make significant progress in this direction by extending the Upper Confidence Bound (UCB) (Auer, Cesa-Bianchi, and Fischer 2002) to deal with the exploration-exploitation tradeoff and employing distributed optimization techniques to properly handle collisions in the presence of walking arms. This require careful integration of these techniques since the default optimal methods are incompatible with external randomness (Vernade, Cappé, and Perchet 2017; Lykouris, Mirrokni, and Paes Leme 2018; Madhushani et al. 2021).

Specifically, we study a "networked information sharing" setting, where all players are arranged in a network $\mathcal{G} := \{\mathcal{N}, \mathcal{E}\}$, and each player has limited capacity for sharing information, e.g., their estimates of the arms' mean rewards with her neighbors in $\mathcal{G}$, as inspired by the original idea of utilizing collisions to share sampled arm rewards in MPMAB settings (Boursier and Perchet 2019; Shi et al. 2020). To tackle the new dilemma in the presence of walking arms, we present a decentralized algorithm called MPMAB-WA-UCB, which is able to avoid collisions after sufficient exploration, in a decentralized manner, i.e., each player decides which arm to pull independently based on the local available information: the past observed rewards and collisions, along with the received information from neighbor players. To achieve this, our high-level idea is to leverage shared information into exploitation to maximize reward from each player's perspective, which turns out to be a matching problem whose complexity grows exponentially with the number of players and arms. To this end, we propose an efficient matching policy and a ranking policy, which assign different rankings to neighbor players so as to avoid collisions. We rigorously prove that a logarithmic growth of the regret is achievable for MPMAB-WA-UCB. Note that our regret analysis is more challenging as traditional regret analysis becomes non-applicable here due to the integration of decentralized optimization methods for handling walking arms.

Related Work

As motivated by the cognitive radio channel assignment problem (Jouini et al. 2009), the MPMAB problems have been extensively studied in different settings. There are two classes of algorithms for MPMAB. The first class allows no information sharing among players, where players sense the presence of other players through experienced collisions (Anandkumar et al. 2011). The other class allows information sharing among players, e.g., directly sharing estimated mean rewards of arms (Liu and Zhao 2010b; Kalathil, Nayyar, and Jain 2014; Rosenski, Shamir, and Szlak 2016; Bistritz and Leshem 2018; Besson and Kaufmann 2018; Boursier and Perchet 2019; Mehrabian et al. 2020; Wang et al. 2020; Bubeck et al. 2020; Lugosi and Mehrabian 2021; Hanawal and Darak 2021; Pacchiano, Bartlett, and Jordan 2021; Shi et al. 2020). In particular, the regret guarantees for MPMAB were significantly improved in (Boursier and Perchet 2019) compared to the non-information sharing case. However, the proposed SIC-MMAB needs to know the time horizon in advance and the exchange of reward estimations leading to the number collisions for communication grows large with T. (Wang et al. 2020; Hanawal and Darak 2021; Shi et al. 2021) extended this model to a leader-follower framework with better regret guarantees.

However, all above literature assume that players have full access to all arms at each time while we consider a setting where players can only access a local subset of arms. Furthermore, the local subset of arms is dynamically changing over time, which exhibits external randomness. As a result, information sharing is necessary for MPMAB-WA to guarantee a near-optimal performance. This is quite intuitive since there exists no universal ranking over arms across players due to the dynamic nature of MPMAB-WA. This results in infinitely often collisions with an $\tilde{\mathcal{O}}(T)$ regret when all players independently pull arms in a greedy way. We provide an intuitive example for further illustration along with additional related work discussions in (Xiong and Li 2022).

Problem Formulation

We consider a stochastic multi-player multi-armed walking bandits (MPMAB-WA) with collisions setting with a set of $\mathcal{N} = \{1, \cdots, N\}$ players, which are randomly distributed in a geographical region, and a set of $\mathcal{K} = \{1, \cdots, K\}$ arms. Each arm k is associated with a reward $X_k(t)$ at each discrete time $t = 1, \cdots, T$. The reward is a random variable on $(0, 1]$ drawn independent and identically distributed (i.i.d.) from a certain distribution associated with arm k with an unknown mean μ_k. Without loss of generalization (W.l.o.g.), we assume that $\mu_1 > \mu_2 > \cdots > \mu_K$. In addition, in real-world applications, each player often has limited capability

for information sharing, e.g., due to limited communication bandwidth. Thus we consider a networked setting where all players are arranged in a connected *communication graph* $\mathcal{G} := \{\mathcal{N}, \mathcal{E}\}$ as the vertices. Denote the neighbor players of player i as $\mathcal{N}_i = \{j|(i,j) \in \mathcal{E}\} \cup \{i\}$.

Walking Arms with Collisions. An arm could sometimes be "walking" and not accessible to a player. Hence we call the set of $\mathcal{K}$ arms as *"walking arms"*. Let $\mathcal{S}_i(t) \subseteq \mathcal{K}$ denote the subset of available arms at time t to player $i \in \mathcal{N}$. We refer to $\mathcal{S}_i(t)$ as *"the local walking arm set"*, satisfying $\bigcup_i \mathcal{S}_i(t) = \mathcal{K}, \forall t$. Since arms are walking, e.g., in a geographical area where players are located at (see our motivating examples in Introduction), we further assume that each arm can only be simultaneously accessed by neighbor players but not disjoint players in $\mathcal{G}$, i.e., $\mathcal{S}_i(t) \cap \mathcal{S}_j(t) = \phi$ if $j \notin \mathcal{N}_i$. At time t, player i can *only* pull an arm from $\mathcal{S}_i(t)$, and *only* observe a non-zero reward[1] if no other neighbor players pull the same arm. Since our reward support is defined on $(0, 1]$, i.e. $\mathbb{P}(X_k = 0) = 0$, the feedback scenarios referred to as *collision sensing* and *no sensing* settings in (Boursier and Perchet 2019) are equivalent.

Networked Information Sharing. Inspired by the original idea of utilizing collisions to share sampled arm rewards (Boursier and Perchet 2019; Shi et al. 2020), each player $i \in \mathcal{N}$ in our `MPMAB-WA` is able to share its local estimates of the arms' mean rewards with her neighbor players in $\mathcal{N}_i$ at each time t. Since players only have access to local walking arm sets in `MPMAB-WA`, and hence there exists no universal ranking over arms across players at each time. Therefore, we further allow each player to share her local walking arm set with her neighbor players.

Policy. A policy π determines which arm each player will pull in each time. We are interested in decentralized policies, where each player determines which arm to pull independently based on the available information to the player, including the past observed collisions, rewards, as well as possible information collected from neighbor players on local walking arm sets and reward estimates. We denote the arm pulled by player i at time t as $a_i(t)$ under policy π, satisfying $a_i(t) \in \mathcal{S}_i(t)$.

Regret. We consider the performance measure of regret (in expectation) incurred by the set of $\mathcal{N}$ players by pulling suboptimal arms under policy π up to time T. Since the local walking arm set $\mathcal{S}_i(t) \subseteq \mathcal{K}, \forall i$ is varying over time t and each player i does not have full access to all arms in $\mathcal{K}$, the optimal arms pulled by all players under the genie-aided algorithm that has knowledge of the true mean reward is not fixed. This differs from existing works where the optimal expected reward can be simply achieved by pulling the best N arms (Rosenski, Shamir, and Szlak 2016; Besson and Kaufmann 2018; Wang et al. 2020). To this end, we denote the actions taken by all players under the genie-aided policy as $\mathbf{a}^*(t) := [a_1^*(t), \ldots, a_N^*(t)], \forall t$, satisfying

$$\mathbf{a}^*(t) = \arg\max_{\{\forall i: a_i(t) \in \mathcal{S}_i(t)\}} \sum_{i=1}^{N} \mu_{a_i(t)} \mathbb{1}_{\{a_i(t) \neq a_j(t), \forall j \neq i, j \in \mathcal{N}\}}, \quad (1)$$

and the corresponding optimal expected reward as $R^* \triangleq \sum_{t=1}^{T} \sum_{i=1}^{N} \mu_{a_i^*(t)}(t)$. Then the regret up to time T of policy π is defined as

$$R(T) \triangleq R^* - \mathbb{E}\left[\sum_{t=1}^{T} \sum_{i=1}^{N} X_{a_i(t)}(t) \mathbb{1}_{\{a_i(t) \neq a_j(t), \forall j \neq i\}}\right]. \quad (2)$$

Remark 1. *The key difference of regret definition in* (2) *with that under full arm access setting (i.e., static arm setting) in prior works is the definition of R^*. Specifically, for a collision-free scenario (Martínez-Rubio, Kanade, and Rebeschini 2019; Madhushani et al. 2021), the N players pull the best arm simultaneously and thus $R^* = TN\mu_1$. For a collision setting (Anandkumar et al. 2011; Boursier and Perchet 2019; Wang et al. 2020), the genie-aided algorithm assigns one of the N-best arms to each player and thus $R^* = T\sum_{i=1}^{N} \mu_i$. The dynamic nature of our `MPMAB-WA` with local walking arm sets for each player brings external randomness and hence renders higher uncertainty for exploration and exploitation. We will discuss its impact on the algorithm design and regret analysis in subsequent sections.*

The `MPMAB-WA-UCB` Algorithm

In this section, we consider `MPMAB-WA` under the above networked information sharing setting, and propose the `MPMAB-WA-UCB` algorithm to address the new dilemma faced by `MPMAB-WA` due to walking arms.

Algorithm Overview

Each player needs to resolve a tradeoff between exploration-exploitation and avoid collisions when attempting to maximize the reward: (i) pulling the arm with the largest estimated reward in her local walking arm set may contribute more to the total reward; and (ii) the neighbor players may share a similar estimation and local walking arm set, which may lead to a collision, and hence degrade the performance. Exacerbating this dilemma is the fact that each player receives feedback from a dynamic local subset of arms at each time. To resolve this dilemma, we leverage the shared information into the exploitation process to avoid collisions while maximizing the reward. At each time t, `MPMAB-WA-UCB` starts with an information sharing process where player i obtains the local walking arm sets $\{\mathcal{S}_m(t), \forall m \in \mathcal{N}\}$, and the local reward estimations $\tilde{r}_{i,k}(t), \forall k \in \mathcal{K}$, from her neighbor players $\forall j \in \mathcal{N}_i$. Then `MPMAB-WA-UCB` alternates between exploration and exploitation as usual based on the past observed collisions and rewards.

Information Sharing. At each time t, player i shares her local estimate of the mean reward $\tilde{r}_{i,k}(t), \forall k \in \mathcal{K}$ with her neighbor players $\forall j \in \mathcal{N}_i$. Meanwhile, player i receives the local estimates from her neighbors in $\mathcal{N}_i$ and then updates her local reward estimates as follows:

$$\tilde{r}_{i,k}(t+1) = \sum_{j \in \mathcal{N}_i} \tilde{r}_{j,k}(t) P_{i,j} + \hat{\mu}_{i,k}(t+1) - \hat{\mu}_{i,k}(t), \quad (3)$$

[1] There are other reward models for `MPMAB` settings, e.g., players can receive a degraded reward, or a full reward is only assigned to one player when collisions occur (Liu and Zhao 2010b,a). In this paper, we assume zero reward (Anandkumar et al. 2011; Besson and Kaufmann 2018) under collision for simplicity. However, our proposed model and algorithm can be easily generalized to other reward settings.

where $\mathbf{P} = (P_{i,j})$ is a $N \times N$ non-negative matrix on the communication graph $\mathcal{G}$ with $P_{i,j} \in [0,1]$, and $\hat{\mu}_{i,k}(t)$ is the empirical estimation of μ_k for player i at time t, which will be specified later in (5). This update is analogous to the decentralized gradient method for decentralized optimization, where $\mathbf{P}$ is referred to as the *consensus matrix*[2], satisfying

$$\begin{cases} P_{i,i} = 1 - \sum_{j \in \mathcal{N}_i} P_{i,j}, \\ P_{i,j} = \frac{1}{\max\{|\mathcal{N}_i|, |\mathcal{N}_j|\}}, \quad \text{if } j \in \mathcal{N}_i, \\ P_{i,j} = 0, \quad \text{otherwise}, \end{cases} \tag{4}$$

with $\sum_{j=1}^{N} P_{i,j} = \sum_{i=1}^{N} P_{i,j} = 1, \forall i, j$. In other words, at each time t, player $i \in \mathcal{N}$ computes a weighted average of the reward estimates of her neighbor players, and then corrects it by taking into account a stochastic approximation $\hat{\mu}_i(t+1) - \hat{\mu}_i(t)$ of her local reward estimate at time t. As aforementioned, each player i also shares her $\mathcal{S}_i(t)$ so as to reach a consensus on the information of local walking arms set of the system, i.e., $\{\mathcal{S}_m(t), \forall m \in \mathcal{N}\}$ at each time t.

Exploration. The exploration of `MPMAB-WA-UCB` is based on the UCB exploration using all observations for each arm inside of the local walking arm set. Essentially, each player runs UCB using the cumulative set of observations it has received. We denote the number of times that player i pulls arms k by time t as $I_{i,k}(t)$, in which collisions occur for $C_{i,k}(t)$ times. Let $X_{i,k}(t)$ be the random reward received by player i when pulling arm k at time t. Then the local reward estimation of $\mu_k, \forall k \in \mathcal{K}$ for player i at time t is given as

$$\hat{\mu}_{i,k}(t) = \frac{\sum_{\tau=1}^{t} \mathbb{1}_{\{a_i(\tau)=k, a_j(\tau)\neq k, \forall j \neq i\}} X_{i,k}(\tau)}{I_{i,k}(t) - C_{i,k}(t)}, \tag{5}$$

where the numerator indicates the total rewards obtained by pulling arm k without collisions, and the denominator denotes the corresponding times that no collisions occur. To accommodate the uncertainty of the local reward estimation $\tilde{r}_{i,k}(t)$ and follow the idea of UCB, we add a perturbed term to the estimated local reward in (3) and define

$$q_{i,k}(t) = \tilde{r}_{i,k}(t) + B_{i,k}(t), \tag{6}$$

with $B_{i,k}(t)$ being a function of $I_{i,k}(t)$ and $C_{i,k}(t)$.

Remark 2. *Player i often regards $q_{i,k}(t)$ in (6) as an index of arm k, and pulls the arm with the largest index at time t for exploitation in most prior works (Anandkumar et al. 2011; Boursier and Perchet 2019; Wang et al. 2020). However, this will inevitably cause a larger number of collisions since the local walking arm sets of neighbor players may share the same arm with the largest estimated reward. To alleviate collisions, learning-to-rank (Combes et al. 2015; Tibrewal et al. 2019) or leader-follower (Wang et al. 2020; Mehrabian et al. 2020) frameworks have been proposed where parsimonious exploration can be done by a single player (i.e., the leader) to find the best N empirical arms, and then send this information to all other players (i.e., the followers). However, these frameworks are based on the assumption that each player has full access to all arms, rendering them inapplicable in `MPMAB-WA`, in which each player only has access to a dynamic local walking arm set. As a result, there exists no such a best empirical arm set accessible for all players. To this end, a new exploitation strategy is needed to leverage the information received from neighbor players in the above information sharing process.*

[2]The easy-to-compute weights in (4) have been widely used in the decentralized optimization literature. Our proposed model and algorithm are not restricted to (4) and can be easily generalized to other stochastic weights for $\mathbf{P}$ (Xiao, Boyd, and Lall 2006).

Algorithm 1: `MPMAB-WA-UCB` for player i at time t

Initialize: The feasible arm sets for each player $\{\mathcal{S}_m(1), \forall m \in \mathcal{N}\}$; the sample mean available at player i $\{\hat{\mu}_{i,k}(1) = 0, \forall k \in \mathcal{K}$, the local estimated reward $\{\tilde{r}_{i,k}(1) = 0, \forall k \in \mathcal{K}$, and the statistics $\{q_{i,k}(1) = \infty, \forall k \in \mathcal{K}\}$; the number of pulls $\{I_{i,k}(1) = 0, \forall k \in \mathcal{K}\}$ and the number of collisions $\{C_{i,k}(1) = 0, \forall k \in \mathcal{K}\}$.

1: **for** $t = 1, ..., T$ **do**
2: Share local walking arm sets among players to yield $\{\mathcal{S}_m(t), \forall m \in \mathcal{N}\}$;
3: Solve (8) by `Learn2Match` and select the arm indicated as $a_i^{*,i}$ by `Learn2Rank`;
4: Update $I_{i,k}(t+1)$ and $C_{i,k}(t+1)$ according to (9);
5: Update $\hat{\mu}_{i,k}(t+1)$ according to (5) and $\tilde{r}_{i,k}(t+1)$ according to (3);
6: Update $q_{i,k}(t+1)$ according to (6).
7: **end for**

Exploitation. After sharing information with neighbor players and estimating the rewards of arms, each player i determines which arm to pull at time t from her local walking arm set $\mathcal{S}_i(t)$. Since we are interested in decentralized decision makings, each player pulls one arm independently based on her local information. As a result, each player i has no information on the selected arms of her neighbor players. To avoid collisions, player i now leverages $\{\mathcal{S}_m(t), \forall m \in \mathcal{N}\}$ along with the estimated reward of $q_{i,k}(t)$ in (6) to determine which arm to pull, instead of simply using $q_{i,k}(t)$ to pull the arm with the largest index value in $\mathcal{S}_i(t)$.

Specifically, let $a_i^m(t)$ be the arm[3] pulled by player $\forall m = 1, \cdots, N$ from the perspective of player i at time t. Denote $\mathbf{a}_i(t) = \left[a_i^1(t), \ldots, a_i^N(t)\right]$ as the set of arms pulled by each player from the perspective of player i, and define the set containing all possible combinations as $\mathcal{U}_i(t)$ satisfying

$$\mathcal{U}_i(t) := \left\{ \mathbf{a}_i(t) \middle| a_i^m(t) \in \mathcal{S}_m(t), \forall m = 1, \cdots, N \right\}. \tag{7}$$

Then player i leverages the collected local walking arm sets $\{\mathcal{S}_m(t), \forall m \in \mathcal{N}\}$, which are now embedded in $\mathcal{U}_i(t)$, together with her local estimated reward $\mathbf{q}_i(t) := \{q_{i,k}(t), \forall k \in \mathcal{K}\}$ to determine which arms all players should pull to maximize reward from her perspective. This

[3]Note that $a_i^m(t)$ is the arm pulled by player m from the perspective of player i, which may not be the true arm pulled by player m since players make decisions in a decentralized manner.

Algorithm 2: `Learn2Match` for player i at time t

Input: $\mathcal{K}$, $\mathbf{q}_i(t), \mathcal{U}_i(t)$.
Ouput: $\mathcal{U}_i^*(t)$.

1: Let $\mathcal{A}_i$ be a permutation on $\mathcal{K}$ with a decreasing order based on the estimated reward $\mathbf{q}_i(t)$, i.e., $q_{i,\mathcal{A}_i^1}(t) \geq q_{i,\mathcal{A}_i^2}(t) \geq \cdots \geq q_{i,\mathcal{A}_i^K}(t)$;
2: **for** $h = 1, 2, ..., N$ **do**
3: Add all players with local walking arm sets containing $\mathcal{A}_i^1$ into $\mathcal{X}_i^h$;
4: **if** $|\bigcup_{l=1}^h \mathcal{X}_i^l| \geq h$ **then**
5: $\mathcal{A}_i = \mathcal{A}_i \setminus \{\mathcal{A}_i^1\}$, $\mathcal{O}_i(t) = \mathcal{O}_i(t) \cup \{\mathcal{A}_i^1\}$;
6: **else**
7: $\mathcal{A}_i = \mathcal{A}_i \setminus \{\mathcal{A}_i^1\}$, $\mathcal{X}_i^h = \phi$, and $h = h - 1$;
8: **end if**
9: **end for**
10: Let $\mathcal{S}_{i,m}^*(t) = \mathcal{S}_m(t) \cap \mathcal{O}_i(t), \forall m \in \mathcal{N}$;
11: Replace $\mathcal{S}_m(t)$ by $\mathcal{S}_{i,m}^*(t), \forall m \in \mathcal{N}$ to obtain $\mathcal{U}_i^*(t)$.

turns out to solving the following matching problem:

$$\max_{\mathbf{a}_i(t) \in \mathcal{U}_i(t)} \sum_{m=1}^{N} q_{i,a_i^m(t)}(t) \mathbb{1}_{\{a_i^m \neq a_i^n, \forall n \neq m, n=1,\cdots,N\}}. \quad (8)$$

Denote the optimal solution to (8) as $\mathbf{a}_i^*(t)$. Then player i pulls arm $a_i^{*,i}(t)$ at time t. Again, we note that $a_i^{*,m}(t)$ is the optimal arm that player $\forall m = 1, \cdots, N$ should pull at time t by solving (8) from the perspective of player i. Finally, player i updates the indicators $I_{i,k}(t+1)$ and $C_{i,k}(t+1)$ based on the outcome of pulling arm $a_i^{*,i}(t)$ at time t, i.e.,

$$I_{i,k}(t+1) = I_{i,k}(t) + \mathbb{1}_{\{a_i(t)=k\}},$$
$$C_{i,k}(t+1) = C_{i,k}(t) + \mathbb{1}_{\{X_{i,k}(t)=0\}}. \quad (9)$$

We summarize our `MPMAB-WA-UCB` algorithm from the perspective of any player $\forall i \in \mathcal{N}$ in Algorithm 1.

`Learn2Match`

To execute the exploration-exploitation process in Algorithm 1, player i needs to solve the optimal matching problem in (8), whose complexity grows exponentially with the number of players N and the number of arms in local walking arm set $\mathcal{S}_m(t), \forall m = 1, \cdots, N$, since $|\mathcal{U}_i(t)| = \prod_{m=1}^N |\mathcal{S}_m(t)|$. To address this challenge, we now develop an efficient matching algorithm named `Learn2Match` to solve (8), which is summarized in Algorithm 2 from the perspective of any player $\forall i \in \mathcal{N}$. Since players receive no rewards when pulling the same arm, our approach to find an optimal $\mathbf{a}_i^*(t)$ to maximize reward from the perspective of player i over all other players is straightforward: based on the local reward estimation $\mathbf{q}_i(t)$ and all players' local walking arm sets $\{\mathcal{S}_m(t), \forall m \in \mathcal{N}\}$, find N "feasible" arms with the largest estimated reward that can be assigned to all players in $\mathcal{N}$ to maximize (8).

Specifically, `Learn2Match` first constructs a permutation on set $\mathcal{K}$, denoted as $\mathcal{A}_i$. W.l.o.g., we order arms in $\mathcal{K}$ in a decreasing order based on the estimated reward $\mathbf{q}_i(t)$, and let $\mathcal{A}_i^k$ denotes the k-th position[4] in $\mathcal{A}_i$ satisfying $q_{i,\mathcal{A}_i^1}(t) \geq q_{i,\mathcal{A}_i^2}(t) \geq \cdots \geq q_{i,\mathcal{A}_i^K}(t)$. Based on this ordering, `Learn2Match` matches arms in $\mathcal{K}$ to all players by checking the arms with estimated rewards in a decreasing order defined by $\mathcal{A}_i$, until finding N feasible arms for all players at time t (lines 2-9 in Algorithm 2). For example, `Learn2Match` first checks the 1st position/arm $\mathcal{A}_i^1$ with the largest estimate reward in $\mathcal{A}_i$, and adds all players whose local walking arm sets contain $\mathcal{A}_i^1$ into $\mathcal{X}_i^1$ (line 3 in Algorithm 2). If the number of such players is no less than $h = 1$, then arm $\mathcal{A}_i^1$ is feasible and should be pulled by one player. Thus `Learn2Match` adds it into the feasible arm set $\mathcal{O}_i(t)$, and removes arm $\mathcal{A}_i^1$ from $\mathcal{A}_i(t)$, i.e., $\mathcal{A}_i = \mathcal{A}_i \setminus \{\mathcal{A}_i^1\}$ (lines 4-5 in Algorithm 2).

Now suppose `Learn2Match` searches for the h-th arm to be added into $\mathcal{O}_i(t)$. `Learn2Match` checks the arm in current $\mathcal{A}_i^1$ and finds all players whose local walking arm sets contain $\mathcal{A}_i^1$ and adds them into $\mathcal{X}_i^h$. If $|\bigcup_{l=1}^h \mathcal{X}_i^l| \geq h$, i.e., the number of players that can pull the h arms in $\mathcal{O}_i(t) \cup \{\mathcal{A}_i^1\}$ is no less than h, and hence `Learn2Match` should remove the current $\mathcal{A}_i^1$ arm from $\mathcal{A}_i$ and put it into its feasible set $\mathcal{O}_i(t)$ (lines 4-5 in Algorithm 2). Otherwise, simple discard this arm since the number of arms in $\mathcal{O}_i(t)$ is enough for all players in $|\bigcup_{l=1}^h \mathcal{X}_i^l|$ to pull (line 7 in Algorithm 2). As a result, `Learn2Match` ends up with a feasible arm set $\mathcal{O}_i(t)$ that contains N unique arms from $\mathcal{K}$ that maximizes (8). Finally, we update the local walking arm sets for all players and obtain the optimal arms pulled by all players at time t as $\mathbf{a}_i^*(t)$ and denote all possibilities as $\mathcal{U}_i^*(t)$ (lines 10-11 in Algorithm 2). The complexity for obtaining $\mathcal{O}_i(t)$ and $\mathcal{U}_i^*(t)$ is linear in the numbers of arms K and players N. Since there may exist more than one optimal arm $a_i^{*,i}(t)$ that all maximize reward over all players from the perspective of player i, we next design a ranking policy named `Learn2Rank` to assign different ranks to all players to determine the unique arm pulled by player i at time t.

Algorithm 3: `Learn2Rank` for player i at time t

Input: $\mathcal{U}_i^*(t)$.

1: Construct $\mathcal{I}_i := \{a_i^{*,i}(t)\}$ and sort $\mathcal{I}_i$ in a decreasing order based on $\mathbf{q}_i(t)$;
2: Construct $\mathcal{J}_i := \{j, j \in \mathcal{N}_i | \mathcal{I}_i \subseteq \mathcal{S}_{i,j}^*(t)\}$;
3: Sort players $\forall j \in \mathcal{J}_i$ in a decreasing order according to their indices;
4: Player i pulls arm $\mathcal{I}_i^{\beta_i}$ with β_i being her ranking.

`Learn2Rank`

Our key observation is that when there are d different optimal arms $a_i^{*,i}(t)$, i.e., $|\mathcal{S}_{i,i}^*(t)| = d$, then there must be d players (including player i herself) that are indifferentiable with these d optimal arms. Let $\mathcal{I}_i := \{a_i^{*,i}(t)\}$ be the set containing all optimal arms that player i can pull at time t. W.l.o.g, we order arms in $\mathcal{I}_i$ in a decreasing order based on

[4]For abuse of notation, $\mathcal{A}_i^k$ refers to the arm in $\mathcal{K}$ with the k-th largest estimated reward from the perspective of player i.

the estimated reward $\mathbf{q}_i(t)$ such that $\mathcal{I}_i^1 \geq \mathcal{I}_i^2 \geq \ldots \geq \mathcal{I}_i^{|\mathcal{I}_i|}$ (line 1 in Algorithm 3). Then, `Learn2Rank` finds the set $\mathcal{J}_i := \{j, j \in \mathcal{N} | \mathcal{I}_i \subseteq \mathcal{S}_{i,j}^*(t)\}$ containing all neighbor players which can pull the optimal arms in $\mathcal{I}_i$ as player i (line 2 in Algorithm 3). In other words, players in $\mathcal{J}_i$ are indifferentiable with arms in $\mathcal{I}_i$. To avoid collisions, a simple rank strategy is to use players' indices. Specifically, `Learn2Rank` sorts players in $\mathcal{J}_i$ in a decreasing order according to their indices, and then player i pulls arm $\mathcal{I}_i^{\beta_i}$ with β_i being the ranking of player i (lines 3-4 in Algorithm 3). This rank assignment associates each player in $\mathcal{J}_i$ with a unique ranking and hence can be used to avoid collisions.

Remark 3. *We note that the idea of ranking players has also been adopted in recent works (Boursier and Perchet 2019; Wang et al. 2020). However, all players are assumed to have full access to all arms at each time. As a result, only one player needs to perform the ranking once and shares the universal ranking with all other players. However, in our* `MPMAB-WA` *model, each player only has access to a local walking arm set that differs across players, and is dynamically changing over time. Hence there exists no universal ranking across players, making existing ranking methods (Boursier and Perchet 2019; Wang et al. 2020) inapplicable. Finally, we provide an example in (Xiong and Li 2022) to illustrate the operations of our proposed* `Learn2Match` *and* `Learn2Rank` *policies.*

Performance Analysis

In this section, we first analyze the performance of our `Learn2Match` and `Learn2Rank` policies, and then provide a finite-time analysis of `MPMAB-WA-UCB`.

Collision Mitigation

We first show that `Learn2Match` and `Learn2Rank` can be used to avoid collisions in `MPMAB-WA`.

Lemma 1. `Learn2Match` *and* `Learn2Rank` *jointly provides an optimal solution to* (8)*, i.e., no collision occurs when the* $\mathbf{q}$*-statistics are accurate.*

Remark 4. *When local reward estimation* $\mathbf{q}$*-statistics at each player are not accurate, players may pull sub-optimal arms and experience collisions, which incur regret (see Theorem 1 and Remark 5). When* $\mathbf{q}$*-statistics are accurate (i.e., after a finite-time of exploration-exploitation), our* `Learn2Match` *and* `Learn2Rank` *jointly ensure an optimal solution to* (8) *without collisions. Our proof consists of two steps. First, based on the construction of* $\mathcal{O}_i(t)$ *in* `Learn2Match` *using the expected estimated reward from the perspective of player* i*, and by contradiction, we show that* $\mathcal{O}_i(t)$ *contains* N *feasible arms, each pulled by one of the* N *players which achieve the largest expected reward for* (8)*. Second, since there may be more than one optimal arm to pull from the perspective of any player* i*, i.e.,* $|\mathcal{S}_{i,i}^*(t)| > 1$*, and players determine which arm to pull in a distributed manner, collisions may occur if each player randomly pull an arm from* $\mathcal{S}_{i,i}^*(t)$*. To this end,* `Learn2Rank` *assigns a ranking to each player to determine the unique arm to pull from* $\mathcal{S}_{i,i}^*(t)$ *and hence avoid collisions.*

Regret Analysis

We now provide a finite-time analysis of `MPMAB-WA-UCB`. For ease of exposition, we define some additional notions. Let $V_{i,k}(t)$ be the number of times that arm $k \in \mathcal{K}$ is only pulled by player $i \in \mathcal{N}$ by time t, and denote $V_k(t) := \sum_{i=1}^N V_{i,k}(t)$. Then the regret defined in (2) reduces to $R(T) = R^* - \sum_{k=1}^K \sum_{i=1}^N \mu_k \mathbb{E}[V_{i,k}(T)]$. Furthermore, we define $I_k(t) := \sum_{i=1}^N I_{i,k}(t)$, where $I_{i,k}(t)$ is the number of times player i pulling arm k by time t as defined earlier. It is straightforward to see that $\sum_{k=1}^K I_k(T) = TN$. We denote $\mathcal{K}_b$ as the set containing arms with the largest N mean reward, i.e., $\mathcal{K}_b := \{\mu_1, \mu_2, \ldots, \mu_N\}$, and let $\mathcal{K}_{-b} = \mathcal{K} \setminus \mathcal{K}_b$ contain the remaining arms. Finally, let $C(T) := \sum_{k \in \mathcal{K}_b} I_k(T) - V_k(T)$ be the number of collisions faced by players by pulling arms in $\mathcal{K}_b$ by time T.

Theorem 1. *The regret of* `MPMAB-WA-UCB` *satisfies* $R(T) \leq$

$$\mu_1 \left(\max \left\{ \sum_{k' \in \mathcal{K}_{-b}} \sum_{k \in \mathcal{K} \setminus \{k'\}} \frac{6 \log T}{(\mu_k - \mu_{k'})^2}, NKL \right\} + \max \left\{ \sum_{k=1}^N \sum_{k'=k+1}^K \frac{6 \log T}{(\mu_k - \mu_{k'})^2}, NKL \right\} + \frac{\pi^2}{3} K(K+N) \right),$$

with $B_{i,k}(t) = \sqrt{\frac{3 \log t}{2 N V_{i,k}(t)}}, \forall i \in \mathcal{N}, k \in \mathcal{K}$ *and* $L = \min_t 3(1-\beta^N)^{t/24N(1+\beta^{-N})} \leq \frac{(1-\beta^N)}{48N(1+\beta^{-N})t}$*, where* β *is the smallest positive value of all consensus matrices, i.e.,* $\beta = \arg\min P_{i,j}$ *with* $P_{i,j} > 0, \forall i, j \in \mathcal{N}$*.*

Remark 5. *The first term corresponds to the regret incurred by pulling suboptimal arms during the exploitation. The second term is incurred by collisions on pulling the best* N *arms when bad rankings caused by incorrect reward estimation, which dominates the regret due to low probability events of bad rankings from our* `Learn2Rank` *policy with good reward estimation. The last term is the regret incurred by the exploration during the initial learning periods, which does not scale with the time horizon* T *since after a finite time of exploration, all players learn the exact rank through our* `Learn2Rank` *policy and hence there would be no regret accumulating afterwards.*

The regret of the first two terms scale with $\mathcal{O}(K^2 \log T)$ and $\mathcal{O}(NK \log T)$, which is sub-logarithmic in time T and matches the regret in existing works, e.g. (Anandkumar et al. 2011; Besson and Kaufmann 2018; Boursier and Perchet 2019; Wang et al. 2020; Mehrabian et al. 2020), where all players are required to have full access to all arms in each time. In contrast each player in our `MPMAB-WA` has the flexibility to access a dynamic subset of arms. Though such flexibility of arm subsets regularly brings external randomness, it does not result in the multiplicative pre-factor that goes with the time-dependent function in the regret to be higher than KN in (Wang et al. 2020; Boursier and Perchet 2019). For instance, the state-of-the-art algorithm SIC-MMAB (Boursier and Perchet 2019) achieves an asymptotically optimal regret of $\tilde{\mathcal{O}}(KN \log T)$ under the

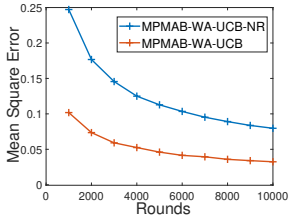

Figure 1: MSE of mean reward.

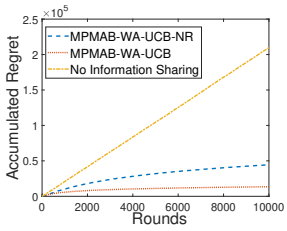

Figure 2: Accumulated regret.

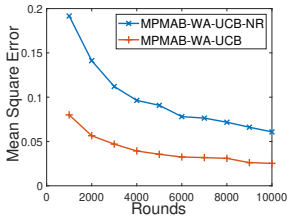

Figure 3: MSE of mean reward in wireless downlink scheduling.

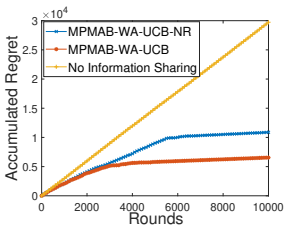

Figure 4: Accumulated regret in wireless downlink scheduling.

assumption that players have full arm access at each time. In addition, it needs to know the time horizon in advance while our MPMAB-WA-UCB requires no knowledge on problem parameters. The number of communication bits is upper bounded by $\mathcal{O}(N^2KT)$. When the network is large, the communication may be predominant over the $\log T$, and hence it is interesting to further explore the joint effect of K and T instead of only considering asymptotic results in T, which largely remains exclusive in multi-player multi-armed bandit settings (Boursier and Perchet 2019).

Remark 6. *As discussed in Related Work, information sharing, in particular, the local walking arm sets, is necessary to guarantee a near-optimal performance for MPMAB-WA since players in MPMAB-WA can only access a local subset of arms, which is also dynamically changing over time. In addition, we allow players to share their local estimates of the arms' mean rewards with their neighbor players in our MPMAB-WA-UCB algorithm as motivated by (Boursier and Perchet 2019; Shi et al. 2020) which showed that such reward estimate sharing in the traditional MPMAB model improved regret guarantees compared to non-sharing case. We now show that this is also true for MPMAB-WA model. Specifically, we consider a variant of MPMAB-WA-UCB, where no reward estimate is shared among players, and call the corresponding policy as MPMAB-WA-UCB-NR. We provide the detailed description of MPMAB-WA-UCB-NR and its regret analysis in (Xiong and Li 2022). As expected, MPMAB-WA-UCB attains an improved regret bound with a factor of $\mathcal{O}(N)$ compared to that of MPMAB-WA-UCB-NR. This is intuitive since player i in MPMAB-WA-UCB also receives the reward estimation from her neighbors $\mathcal{N}_i$ at each time, where $|\mathcal{N}_i| < N$, which can be regarded as a means to improve the exploration efficiency by a factor of $\mathcal{O}(N)$, i.e., an $\mathcal{O}(N)$ decrease for the number of time steps needed to obtain the accurate statistics of arms.*

Numerical Evaluations

Experiments on Constructed Instance. We consider $N = 6$ players and $K = 100$ arms with rewards drawn from Gaussian distributions with mean $\mu_k = 0.06(101 - k)$ and $\sigma_k = 0.01(101 - k)$. Each player has three neighbor players in the communication graph $\mathcal{G}$. At each time, we randomly assign 25 arms to each player with neighbor players possibly sharing some arms. All the regret and MSE values are averaged over 40 independent runs. Figure 1 compares the mean-square-error (MSE) between each arm's true mean reward and estimated mean reward with our proposed algorithms over a time horizon of $T = 10^4$ rounds. It is clear that sharing estimated rewards with neighbor players as in MPMAB-WA-UCB substantially improves the exploration efficiency compared to only sharing local walking arm sets as in MPMAB-WA-UCB-NR. This advantage results in a lower regret as shown in Figure 2, which is consistent with our theoretical performance guarantees. Finally, we observe that communication significantly improves the performance since communication is required to determine optimal matching and ranking to avoid collisions. Its importance is especially pronounced when players only have access to a dynamic local walking arm set as considered in this paper.

Experiments on Wireless Downlink Scheduling. We further consider a wireless downlink scheduling problem (Li 2021; Li, Liu, and Ji 2019) that fits into our MPMAB-WA model, see (Xiong and Li 2022) for details. There are $N = 6$ base stations (BSs) and $K = 10$ walking users. Each BS covers a geographical region and each user randomly moves across the whole region with uniform distribution, i.e., each user moves into the region covered by BS n with a probability $1/N$ at each time slot. BSs are connected via a ring, i.e., each BS has two neighbors. The rewards of serving users in each slot (Huang, Hu, and Pan 2021) are i.i.d. drawn from Bernoulli distributions with mean rewards $0.95, 0.9, 0.85, 0.8, 0.75, 0.7, 0.65, 0.6, 0.55, 0.5$. All MSE and regret reported in Figures 3 and 4 are averaged over 40 independent runs, from which we draw the same conclusions as above.

Conclusion

In this paper, we studied the stochastic multi-player multi-armed bandits with collisions problem in the presence of walking arms, dubbed as MPMAB-WA. This new framework integrates several critical factors of systems for many real-world applications. In MPMAB-WA, each player only has access to a dynamic local walking arm set at each time, and only observes a full reward if no other players pull the same arm. This introduced a new dilemma to manage the balance between maximizing the reward via exploration-exploitation, and avoiding collisions when players only receive feedback from a dynamic local walking arm set. To address this challenge, we considered a practical information sharing setting to coordinate players, and proposed a decentralized algorithm with theoretical guarantee on the regret.

Acknowledgements

This work was supported in part by the National Science Foundation (NSF) grants CRII-CNS-NeTS-2104880 and RINGS-2148309, and was supported in part by funds from OUSD R&E, NIST, and industry partners as specified in the Resilient & Intelligent NextG Systems (RINGS) program, as well as the U.S. Department of Energy (DOE) grant DE-EE0009341. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

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.

Andrews, J. G.; Buzzi, S.; Choi, W.; Hanly, S. V.; Lozano, A.; Soong, A. C.; and Zhang, J. C. 2014. What will 5G be? *IEEE Journal on Selected Areas in Communications*, 32(6): 1065–1082.

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

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

Bistritz, I.; and Leshem, A. 2018. Distributed multi-player bandits-a game of thrones approach. *Advances in Neural Information Processing Systems*, 31.

Boursier, E.; and Perchet, V. 2019. SIC-MMAB: Synchronisation Involves Communication in Multiplayer Multi-Armed Bandits. *Advances in Neural Information Processing Systems*, 32: 12071–12080.

Bubeck, S.; and Cesa-Bianchi, N. 2012. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Machine Learning*, 5(1): 1–122.

Bubeck, S.; Li, Y.; Peres, Y.; and Sellke, M. 2020. Non-stochastic multi-player multi-armed bandits: Optimal rate with collision information, sublinear without. In *Conference on Learning Theory*, 961–987. PMLR.

Ceselli, A.; Premoli, M.; and Secci, S. 2017. Mobile edge cloud network design optimization. *IEEE/ACM Transactions on Networking*, 25(3): 1818–1831.

Combes, R.; Magureanu, S.; Proutiere, A.; and Laroche, C. 2015. Learning to rank: Regret lower bounds and efficient algorithms. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, 231–244.

Farhadi, V.; Mehmeti, F.; He, T.; La Porta, T. F.; Khamfroush, H.; Wang, S.; Chan, K. S.; and Poularakis, K. 2021. Service placement and request scheduling for data-intensive applications in edge clouds. *IEEE/ACM Transactions on Networking*, 29(2): 779–792.

Gupta, S.; Chaudhari, S.; Joshi, G.; and Yağan, O. 2021. Multi-armed bandits with correlated arms. *IEEE Transactions on Information Theory*.

Hanawal, M. K.; and Darak, S. 2021. Multi-player bandits: A trekking approach. *IEEE Transactions on Automatic Control*.

Huang, Z.; Hu, B.; and Pan, J. 2021. Poster: Multi-agent Combinatorial Bandits with Moving Arms. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, 1140–1141. IEEE.

Jouini, W.; Ernst, D.; Moy, C.; and Palicot, J. 2009. Multi-armed bandit based policies for cognitive radio's decision making issues. In *2009 3rd International Conference on Signals, Circuits and Systems (SCS)*, 1–6. IEEE.

Kalathil, D.; Nayyar, N.; and Jain, R. 2014. Decentralized learning for multiplayer multiarmed bandits. *IEEE Transactions on Information Theory*, 60(4): 2331–2345.

Lai, T. L.; and Robbins, H. 1985. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1): 4–22.

Landgren, P.; Srivastava, V.; and Leonard, N. E. 2016. Distributed cooperative decision-making in multiarmed bandits: Frequentist and bayesian algorithms. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, 167–172. IEEE.

Li, B. 2021. Efficient learning-based scheduling for information freshness in wireless networks. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, 1–10. IEEE.

Li, F.; Liu, J.; and Ji, B. 2019. Combinatorial sleeping bandits with fairness constraints. *IEEE Transactions on Network Science and Engineering*, 7(3): 1799–1813.

Liu, K.; and Zhao, Q. 2010a. Decentralized multi-armed bandit with multiple distributed players. In *2010 Information Theory and Applications Workshop (ITA)*, 1–10. IEEE.

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

Lugosi, G.; and Mehrabian, A. 2021. Multiplayer bandits without observing collision information. *Mathematics of Operations Research*.

Lykouris, T.; Mirrokni, V.; and Paes Leme, R. 2018. Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 114–122.

Madhushani, U.; Dubey, A.; Leonard, N.; and Pentland, A. 2021. One more step towards reality: Cooperative bandits with imperfect communication. *Advances in Neural Information Processing Systems*, 34.

Martínez-Rubio, D.; Kanade, V.; and Rebeschini, P. 2019. Decentralized cooperative stochastic bandits. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 4529–4540.

Mehrabian, A.; Boursier, E.; Kaufmann, E.; and Perchet, V. 2020. A practical algorithm for multiplayer bandits when arm means vary among players. In *International Conference on Artificial Intelligence and Statistics*, 1211–1221. PMLR.

Pacchiano, A.; Bartlett, P.; and Jordan, M. I. 2021. An Instance-Dependent Analysis for the Cooperative

Multi-Player Multi-Armed Bandit. *arXiv preprint arXiv:2111.04873*.

Robbins, H. 1952. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5): 527–535.

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

Shi, C.; Xiong, W.; Shen, C.; and Yang, J. 2020. Decentralized multi-player multi-armed bandits with no collision information. In *International Conference on Artificial Intelligence and Statistics*, 1519–1528. PMLR.

Shi, C.; Xiong, W.; Shen, C.; and Yang, J. 2021. Heterogeneous Multi-player Multi-armed Bandits: Closing the Gap and Generalization. *Advances in Neural Information Processing Systems*, 34.

Tibrewal, H.; Patchala, S.; Hanawal, M. K.; and Darak, S. J. 2019. Multiplayer multi-armed bandits for optimal assignment in heterogeneous networks. *arXiv preprint arXiv:1901.03868*.

Vernade, C.; Cappé, O.; and Perchet, V. 2017. Stochastic Bandit Models for Delayed Conversions. In *Conference on Uncertainty in Artificial Intelligence*.

Wang, P.-A.; Proutiere, A.; Ariu, K.; Jedra, Y.; and Russo, A. 2020. Optimal algorithms for multiplayer multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics*, 4120–4129. PMLR.

Xiao, L.; Boyd, S.; and Lall, S. 2006. Distributed Average Consensus with Time-Varying Metropolis Weights. *Automatica*.

Xiong, G.; and Li, J. 2022. Decentralized Stochastic Multi-Player Multi-Armed Walking Bandits. *arXiv preprint arXiv:2212.06279*.