Distributed Algorithms for Efficient Learning and Coordination in Ad Hoc Networks

Arun Verma and Manjesh K. Hanawal
Industrial Engineering and Operations Research
Indian Institute of Technology Bombay
{v.arun,mhanawal}@iitb.ac.in

Rahul Vaze
School of Technology and Computer Science
Tata Institute of Fundamental Research
vaze@tcs.tifr.res.in

***Abstract*—A distributed sampling strategy for multiple (N) agents is considered that minimizes the sample complexity and regret of acquiring the best subset of size N among total $K \geq N$ channels in a cognitive radio access setup. Agents cannot directly communicate with each other, and no central coordination is possible. Each agent can transmit on one channel at a time, and if multiple agents transmit on the same channel at the same time, a collision occurs, and no agent gets any information about the channel gain or how many other agents transmitted on the same channel. If no collision occurs, the agent observes a reward (or gain) sample drawn from an underlying distribution associated with the channel. An algorithm to minimize the sample complexity and regret is proposed. One important property of our algorithm that distinguishes it from the prior work (that do not assume knowledge of N) is that it requires no information about the difference of the means of the channel gains of the K channels. Our approach results in fewer collisions with improved regret performance compared to the state-of-the-art algorithms. We validate our theoretical guarantees with experiments.**

***Index Terms*—Multi-player Multi-armed Bandits, Constant regret, Pure exploration, Distributed learning**

I. Introduction

Consider a communication network consisting of N users and K channels where each user wishes to use one of the K channels. A user can transmit on any of the channels. To keep the model more general, users cannot communicate directly between them and any information about the other users can be obtained only by collisions. In particular, if user n transmits on channel k, it gets a sample of the gain of channel k if no other user transmits on channel k, otherwise it only gets to know that at least one other user transmitted on channel k at the same time. We assume that the expected gain of each of the K channels is distinct. Even though users cannot communicate and have to make decisions in a distributed fashion, we consider a non-strategic setting where users have a common goal to maximize the total gain in the network, i.e., to achieve an optimal allocation, which is realized when all the users occupy non-overlapping channels in the top N channels. Here the top N channels refer to the set of N channels with highest expected gain. As the goal is to achieve optimal network allocation, we assume that each user is satisfied if she gets one of the top N channels among the K, when no other better channel is free. Our aim in this work is to find a strategy that minimizes the time by which the users reach an optimal allocation while keeping the number of collisions low.

This problem is motivated by ad hoc cognitive radio networks (CRN), where multiple users try to access the same set of channels [1], [2], without any direct communication between them. In this setting, there is no central controller or a common control channel that can be used to resolve contentions and all channel selection decisions have to be done in a decentralized fashion [1], [2]. Moreover, this problem is well suited for upcoming $5G$ standards like device-to-device communication mode in cellular networks where multiple mobile nodes try to form groups to reduce signaling load on the central base station. Such models are also being envisioned for futuristic ultra-dense networks deployed to offer high peak rates [3].

In ad hoc CRNs, users may not know the quality of channels, and the number of other users present in the network. The goal is still to maximize the number of successful transmissions (or sum rate/ throughput) in the network. To achieve this, the users not only have to learn the channel qualities but also have to learn to co-ordinate by selecting non-overlapping channels. This distributed learning problem is a multi-player version of the multi-armed bandit problem with K arms, with a total of N players. Individual players do not know the number of other players and cannot communicate with others about their arm selection strategy. A player can select at most one arm at a time. If multiple players select a common arm, a collision occurs and none of them receive any reward.

The performance of the arm selection policy is measured as regret, i.e., the difference of the total expected reward obtained by the policy and the total expected reward gained by playing an optimal strategy in each round by all the players. The total number of collisions is the sum of collisions experienced by all the players. From the applications point of view, e.g., in a CRN, where users are mostly battery operated, a higher number of collisions will result in reduced operational life. Hence, in addition to minimizing the regret, the algorithms should work with as fewer collisions as possible.

The problem of multi-player multi-armed bandits with the unknown number of users is studied in [4] where Musical chair (MC) is developed to minimize regret and its performance is shown to be superior compared to other algorithms (for the unknown number of players) [5] [6]. In the MC algorithm, each player selects the arms uniformly at random for a fixed number of rounds and then estimate the top N arms and the number of users based on the reward samples and collisions observed. The number of rounds required to get good estimates is set based on the separation between the mean reward of the arms and is assumed to be lower bounded by a known positive number. However, this assumption is restrictive as it is not easy to identify the lower bound in real-life applications. In this work, we propose a learning algorithm which achieves the optimal allocation quickly with fewer collision without requiring the knowledge of the gap between the mean reward of the arms. Our contributions can be summarized as follows:

- We design a communication protocol for exchanging information among players through collisions that do not need any direct communication medium between players.

- We develop an algorithm named Distributed Learning and Coordination (DLC) that achieves optimal allocation in the constant number of the rounds with high probability. This bound also translates to a constant regret with high probability. The DLC algorithm does not need to know the minimum gap between the arms.
- We give a variant of DLC that improves utilization of channels through efficient signaling, resulting in performance improvement in terms of regret.
- We validate our claims by numerical experiments and demonstrate performance gains over the state-of-the-art.

The DLC algorithm combines the ideas of pure exploration methods with efficient signaling schemes to achieve optimal allocation with high probability. In DLC, each player first learns the number of players in the network by colliding with others in a specific pattern. One of the players then identifies the top N arms via pure explorations and signals all the players to occupy one of top N arms without any overlap.

A. Related Work: Multi-player Multi-armed Bandits

Most work on stochastic bandits with multiple-players require some negotiation or pre-agreement phase to avoid collisions between the players. The dUCB$_4$ algorithm in [7] achieves this using Bertsekas' auction mechanism for players to negotiate unique arm. The time divisions fair sharing (TDFS) algorithm in [8] requires players to agree on a time division of slots before the game. Such negotiations are hard to realize in a completely distributed (ad hoc) setup [1], [2]. The ρ^{RAND} algorithm in [5] is communication free and completely decentralized but requires knowledge of the number of users. The modified ρ^{EST} algorithm overcomes this issue, but its guarantees are asymptotic and do not hold for the finite time like ours. Performance improvements of ρ^{RAND} are studied in [9]. Another set of works in [10], [11] considers selfish behavior of players and analyze their equilibrium behavior.

The works most similar to ours are [6] and [4] which consider a communication-free setting with the unknown number of players that can vary during the game. The MEGA algorithm in [6] uses the classical ϵ- greedy MAB algorithm and ALOHA based collision avoidance mechanism. Though collision frequency reduces in MEGA as the game proceeds it may not go to zero as shown in [4]. To overcome this, [4] proposed a musical chairs (MC) algorithm that incurs collisions only in the initial phase and guarantees collision free sampling subsequently. Though MC performs better than MEGA, its performance in the initial rounds is poor – MC uses collision information to estimate the number of players and forces a large number of collisions to get a good estimate.

SIC-MMAB and ESER algorithm in [12] and [13], respectively, propose signaling mechanisms that use a suitable pattern of collisions among the players for exchanging information. Although some of the ideas of the DLC and SIC-MMAB are similar, DLC uses different learning and signaling scheme. Specifically, the exploration phase in SIC-MMAB and ESER is based on sequential hopping whereas in DLC it is based on pure-exploration.

Organization of the paper: In Section II we introduce the notations and setup the problem. We give an algorithm and analyze their performance in Section III and provide its improved variant in Section IV. We validate our claims through experiments in Section V. Conclusions and future directions are given in Section VI.

II. Problem Setup

The standard stochastic K-armed bandit problem consists of a single player with $K > 1$ arms. The multi-player K-armed bandit is similar, but consists of multiple players. Let $N \leq K$ denote the total number of players. Playing arm $k \in [K]$ where $[K] := \{1, 2, \ldots, K\}$, gives reward drawn independently from a distribution with support $[0,\ 1]$. The reward distributions are stationary, homogeneous, and independent across the players, and μ_k denotes the mean of arm k. The players are not aware of how many other players are present, and there exist no control channels over which they can communicate with each other. When a player samples an arm, the reward is obtained if she alone happens to play that arm. Otherwise, all the players choosing that arm will get zero rewards. We refer to the latter case as 'collision'. For any distributed policy, let $I_{n,t}$ and $\eta_{n,t}$ indicate the arm played by player $n \in [N]$ and her collision indicator in the round t, respectively. $\eta_{n,t} = 1$ if together with player n at least one other player plays the same arm as n at time t and is set equal to zero otherwise.

To achieve an optimal allocation in a distributed setting, each player needs to learn the means of the arms to arbitrary precision which require them to play a large number of times making the goal infeasible for all practical purposes. Thus, we focus on achieving an approximate optimal allocation of arms with high probability that is defined as below. Let $\pi := \pi_t : t \geq 1$ denote a policy, where $\pi_t : [N] \to [K]$ denotes the allocation at time t, i.e., $I_{n,t} = \pi_t(n)$.

Definition 1 (Def. 1 of [13])**.** *For a given tolerance $\epsilon \geq 0$ and confidence $\delta \in (0, 1/2]$, an allocation by a policy π is said to be $(\epsilon, \delta)-$optimal if there exists $T := T(\pi) < \infty$ such that*

$$\Pr\left\{\sum_{n\in[N^\star]} \mu_n - \sum_{n\in[N]} \mu_{\pi_t(n)} \leq \epsilon\right\} \geq 1-\delta. \quad \forall\, t \geq T \quad (1)$$

where $N^\star$ denotes the best subset of N arms with highest mean rewards.

This definition can be viewed as a generalization of probably-approximately-correct (PAC) performance guarantee in the pure exploration multi-armed bandits problems with single player [14], [15] to the multi-player case. $T(\pi)$ denotes the sample complexity of the policy π. The goal is to develop a policy that has small sample complexity.

We define the expected cumulative (pseudo-)regret of policy π over period T as

$$\mathcal{R}_T(\pi) = \sum_{t=1}^{T} \sum_{n\in[N^*]} \mu_n - \sum_{t=1}^{T} \sum_{n\in[N]} \mu_{\pi_t(n)}(1-\eta_{n,t}). \quad (2)$$

The total number of collisions incurred by a policy π over period T is defined as

$$C(\pi) = \sum_{t=1}^{T} \sum_{n\in[N]} \eta_{n,t}. \quad (3)$$

Note that a collision on the arm is counted multiple times because all the players involved in a collision have to re-sample in another slot incurring extra transmission cost.

Our goal is to develop distributed algorithms that gives $(\epsilon, \delta)-$optimal allocation quickly while keeping $\mathcal{R}_T$ and $\mathcal{C}_T$ low. Specifically, we develop algorithms whose regret is constant with high probability, i.e., algorithm incurs regret only for finitely many rounds.

Similar to the prior work [4], [9], [12], [13], [16], we assume players are synchronized and know arms' indices before entering into the network. Further, all the players are assumed to see the same gains on all the channels, which is often the case in dense networks because of the close proximity of users.

III. An Adaptive Algorithm

In this section, we propose an algorithm named, Distributed Learning and Coordination (DLC), in which all the players settle on top N arms quickly. The players cannot communicate explicitly but can exchange information with each other by colliding in a particular fashion. We refer to such deliberate collisions in the network for information exchange as *signals*. All signals are counted as collisions and add to the regret.

A. DLC *Algorithm*

DLC algorithm consists of mainly 4 phases, namely 1) Orthogonalization 2) Player Indexing 3) Adaptive Learning and 4) Communication. These phases run sequentially one after another.

DLC **D**istributed **L**earning and **C**oordination

Input: $K,\ \epsilon \geq 0,\ \delta \in (0, 1/2]$
Select the arms uniformly at random for T_{RP} rounds and find reserved arm i
Play arm i for $2i$ rounds. After that sequentially hop for $K - i$ rounds and then play arm i for $K - i - 1$ rounds
Find number of players (N) and their reserved arm. Designate as Leader if has smallest reserved arm's index
if Player is Leader **then**
    Find top N arms using *AdaptiveExplore* sub-routine
    Assign top N arms among players without overlap
else
    Transmit on arm i periodically. When a collision is observed on arm i, play it for next $\lceil \log_2(K) \rceil$ rounds and then play received arm in the subsequent rounds.
end if

1) Orthogonalization: The first phase of DLC finds orthogonal arm allocations through random hopping in which each player selects an arm uniformly at random in each round until she observes a collision-free transmission on the selected arm. Once it happens, she continues to play that arm till the end of the phase. The length of *Orthogonalization* phase (T_{RP}) is set such that all the players are on different arms by the end of the phase with probability at least $1 - \delta$. We refer to the last arm played by a player in this phase as her *reserved arm*.

2) Player Indexing: This phase is similar to the initialization phase in SIC-MMAB algorithm [12]. In this phase, each player estimates how many other players are there in the network and finds their reserved arms using a specific pattern of collisions. A player with reserved arm i plays arm i for $2i$ rounds. After this, she starts playing arm with higher index in each round (sequential hopping) for next $K - i$ rounds and then plays her reserved arm for next $K - i - 1$ rounds. This process makes sure that players begin sequential hopping in a delayed fashion and the player with reserved arm i will collide with the player with the reserved arm $j (> i)$ at time $T_{RP} + i + j$. At the end of round $T_{RP} + 2K - 1$, each player knows the number of players (N) in the network and their respective reserved arms.

The player with smallest reserved arm's index will be designated as *Leader*. The Leader knows the reserved arm of other players and she assigns a ranking to the players based on the index of their reserved arms as follows: The Leader takes herself rank 0. The player with the second smallest index of the reserved arm is assigned rank 1. The third smallest index of the reserved arm has rank 2 and so on, i.e., the player on j^{th} smallest index of the reserved arm is given rank $j - 1$.

3) Adaptive Learning: In the next phase, the Leader uses the *AdaptiveExplore* sub-routine to find the top N arms while the non-Leaders check for a signal from the Leader by transmitting on their reserved arms periodically. Periodic transmissions of all players in the non-Leaders set are designed such that only one player among the non-Leaders transmits at a time. It helps the Leader to inform the non-Leaders which arms they should occupy when she has completed the task of identifying the top N arms. We first describe a method for the Leader to learn the top N arms and then describe a method how she can inform the Non-Leaders which arms to occupy.

Sub-routine: *AdaptiveExplore*

Input : $K,\ N \leq K,\ \epsilon \geq 0,\ \delta \in (0, 1/2], t = 1$
Initialize: $\alpha = 1.1,\ k_1 = 505.5,\ B(1) = \infty$
Play each arm once. Compute $U_a(1), L_a(1)\ \forall\ a \in [K]$
while $B(t) > \epsilon/N$ **do**
    Compute u_t and l_t using (5) and play arm u_t and l_t in round-robin fashion. If no non-Leader is going to play selected arm then play it else play other arm
    Update $J(t)$, $U_a(t)$ and $L_a(t)\ \forall a \in [K]$ using (4)
    $B(t) \leftarrow U_{u_t} - L_{l_t}, t \leftarrow t + 2$
end while
$T_{AE} \leftarrow t$
Return $J(T_{AE}),\ T_{AE}$

The Leader can use any pure exploration algorithm [14], [15], [17] to find the top N-arms. We will adapt the KL-LUCB algorithm given in [15] to our scenario as it has the best-known performance guarantees.

AdaptiveExplore takes (K, N, ϵ, δ) as input and maintains a set $J(t)$ of the N arms with highest empirical mean rewards in each round t. $U_a(t)$ and $L_a(t)$ represent the upper and lower confidence bounds on a mean reward μ_a of arm a. These bound are computed using empirical mean reward $\hat{\mu}_a(t)$ of arm a at time t as follows:

$$U_a(t) := \max\{q \in [\hat{\mu}_a(t), 1] : N_a(t) d(\hat{\mu}_a(t), q) \leq \beta(t, \delta)\}$$
$$L_a(t) := \max\{q \in [0, \hat{\mu}_a(t)] : N_a(t) d(\hat{\mu}_a(t), q) \leq \beta(t, \delta)\}$$
$$\text{where } \beta(t, \delta) = \log\left(\frac{k_1 K s^\alpha}{\delta}\right) + \log\log\left(\frac{k_1 K s^\alpha}{\delta}\right),$$
$$\alpha > 1, k_1 > 2e + 1 + \frac{e}{\alpha - 1} + \frac{e + 1}{(\alpha - 1)^2}, s = \lceil t/2 \rceil, \quad (4)$$

and $d(p, q)$ is the Kullback-Leibler divergence (differential entropy) between two Bernoulli distributions with parameter p and q, and δ is the fixed confidence. In round t,

AdaptiveExplore selects two arms u_t and l_t such that

$$u_t = \arg\max_{j \notin J(t)} U_j(t) \text{ and } l_t = \arg\min_{j \in J(t)} L_j(t). \quad (5)$$

AdaptiveExplore finds (ϵ, δ)-optimal allocation if it terminates after:

$$\boxed{U_{u_t}(t) - L_{l_t}(t) < \epsilon/N} \quad (6)$$

where $\epsilon \geq 0$ is fixed tolerance. Its proof follows from Theorem 1 of [14] by setting tolerance equal to ϵ/N.

AdaptiveExplore selects two arms in each alternative rounds, and they are played over two slots in a round-robin fashion until stopping criteria in Eqn. (6) is not met. In each round, before playing an arm, the Leader checks that none of the non-Leaders will play that arm in that round. If any non-Leader is going to play the selected arm (due to periodic signals), Leader plays the other arm to avoid any collision. The periodic signaling by the non-leaders is designed such that they play only one arm in any round.

When AdaptiveExplore terminates, it returns the top N arms and number of rounds (T_{AE}) it takes to terminate. Next, Leader does a one-to-one mapping of top N arms to the N players. This mapping can be done randomly or by using an injective function. In our implementation, we assign the top arm to the player with the best rank and the next best arm to the player with next best rank and so on. The Leader takes the N^{th} best arm. After mapping arms to players, Leader informs other players which arm they should occupy using collisions based communication protocol.

Note that mapping of top N arms to players can be easily adapted to be 'fair' allocation amongst players so that all the players have same average reward asymptotically. For example, after the Leader communicates the allocations, they can sequentially hop on the N arms, which ensures no collision and optimal allocation in every round.

4) Communication: While the Leader explores the arms to find the top N arms, others play their reserved arms periodically. Each non-Leader plays her reserved arm based on her rank. A player with better rank plays earlier than the higher ranked players. Specifically, a non-Leader ranked r plays her reserved arm for the m^{th} time in the round $P(r, m)$ given by

$$P(r, m) = mT_P + (r-1)B_K$$
$$\text{where } B_K = (\lceil \log_2(K) \rceil + 1) \text{ and } T_P \geq (N-1)B_K.$$

Once AdaptiveExplore sub-routine is initiated by the Leader, the player with rank 1 plays her reserved arm once after T_P rounds, and then the player with next rank (i.e., 2) plays her reserved arm. The value of T_P is chosen such a way that all players can play their reserved arm and receive arm information from Leader within it, therefore, $T_P \geq (N-1)(\lceil \log_2 K \rceil + 1)$. This signaling strategy enables the Leader to transmit arm information to a player immediately after she finishes it for the previous player and continues until every player gets their mapped arm from the Leader.

After termination of AdaptiveExplore sub-routine, i.e., at time T_{AE}, Leader computes the next time when a player with rank r will play her reserved arm, denoted $P_L(r)$, as $P_L(r) = P(r, \tau)$ if $P(r, \tau) > T_{AE}$, otherwise $P_L(r) = P(r, \tau + 1)$ where $\tau = \lfloor T_{AE}/T_P \rfloor$. When the Leader wants to send arm information to the player with rank r, she will play the reserved arm of that player in round $P_L(r)$. The occurrence of the first collision notifies the player that she is going to get information of the arm to occupy from the Leader and starts playing her reserved arm for $\lceil \log_2 K \rceil$ rounds. As the index of each arm is uniquely represented by the binary number of length $\lceil \log_2 K \rceil$, these many rounds are sufficient to convey the arm index. The Leader informs a non-Leader the index of the arm to occupy by colliding with her according to the binary sequence of the arm's index over $\lceil \log_2 K \rceil$ rounds. Leader conveys bit '1' by causing a collision on her reserved arm, while a bit '0' is conveyed if no collision occurs. With such a collision pattern, all players get to know which arms they should occupy in the subsequent rounds.

B. Analysis of DLC

We prove sample complexity, regret and collision bounds of DLC algorithm using the following lemmas which give expected number of rounds for players to 1) orthogonalize 2) learn the number of players in the system with their ranks 3) find top N arms and 4) arm assignment through signaling.

Theorem 1. *Let arms be ordered in descending order of their mean rewards and set $c = (\mu_N + \mu_{N+1})/2$ where $c \notin \{0, 1\}$. For any $\epsilon \geq 0$ and $\delta \in (0, 1/2]$ the sample complexity of* DLC *is bounded with probability at least $1 - 2\delta$ as*

$$T(DLC) \leq \left\lceil \frac{\log(\delta/K)}{\log\left(1 - \frac{1}{4K}\right)} \right\rceil + 2K - 1 + C_0(\alpha) H_{\epsilon,c} \log\left(\frac{k_1 K (H_{\epsilon,c})^\alpha}{\delta}\right) + N(\lceil log_2(K) + 1 \rceil$$

where $H_{\epsilon,c} := \sum_{a \in [K]} \frac{1}{\max\{d(\mu_a, c), \epsilon^2/2\}}$, $k_1 > 2e + 1 + \frac{e}{\alpha - 1} + \frac{e+1}{(\alpha-1)^2}$ with $\alpha > 1$ and $C_0(\alpha)$ is a problem independent constant. Further, the total collisions in DLC *is bounded as*

$$C(DLC) \leq N \left\lceil \frac{\log(\delta/K)}{\log\left(1 - \frac{1}{4K}\right)} \right\rceil + (N-1)\left(\frac{N}{2} + \lceil \log_2 K \rceil + 1\right).$$

The proof follows by bounding the number of rounds required to complete each phase and the number of collisions incurred in these phases. These bounds are given by the following lemmas whose proofs are deferred to the appendix.

Lemma 1. *Let δ be same as in Theorem 1. All players will orthogonalize with probability at least $1 - \delta$ in Orthogonalization phase within T_{RP} rounds where*

$$T_{RP} := \left\lceil \frac{\log(\delta/K)}{\log\left(1 - \frac{1}{4K}\right)} \right\rceil.$$

The number of collisions in the Orthogonalization phase is at most $C_{RP} \leq NT_{RP}$.

Lemma 2. *In the Player's Indexing phase, each player learns about number of players in the network, their reserved arm, and rank in T_{IP} rounds where $T_{IP} := 2K - 1$. During this phase, the number of collisions is $C_{IP} := N(N-1)/2$.*

Lemma 3. *Assume conditions in Theorem 1 hold. Then with probability at least $1 - \delta$, AdaptiveExplore terminates with (ϵ, δ)-optimal allocation after T_{AE} rounds where*

$$T_{AE} \leq C_0(\alpha) H_{\epsilon,c} \log\left(\frac{k_1 K (H_{\epsilon,c})^\alpha}{\delta}\right)$$

where $\alpha > 1, C_0(\alpha)$ is a problem independent constant satisfying $C_0(\alpha) \geq \alpha \log(C_0(\alpha)) + 1 + \alpha/e, k_1 > 2e + 1 + \frac{e}{\alpha-1} + \frac{e+1}{(\alpha-1)^2}$, and $H_{\epsilon,c} := \sum_{a\in[K]} \frac{1}{\max\{d(\mu_a,c),\epsilon^2/2\}}$.

Proof. KL-LUCB algorithm samples two arms in each round whereas AdaptiveExplore sub-routine samples one arm. Using $t = 2s$ in the Theorem 3 of [15], we achieve stated bound. □

Lemma 4. *Let $T_P = (N-1)(\lceil \log_2 K \rceil + 1)$. Then the number of rounds to terminate the Communication phase (T_{CP}) and the number of collisions incurred (C_{CP}) are bounded as*

$$T_{CP} \leq N(\lceil log_2(K) \rceil + 1),$$
$$C_{CP} \leq (N-1)(\lceil \log_2 K \rceil + 1).$$

We now return to the proof of Theorem 1. $T(DLC)$ is the total rounds from the beginning of Orthogonalization phase to end of Communication phase, i.e.,

$$T(DLC) = T_{RP} + T_{IP} + T_{AE} + T_{CP}. \tag{7}$$

Substituting the bounds on each of these terms from Lemmas 1, 2, 3 and 4 in (7) we get the bound.

Similarly, the total number of collisions is the sum of that occurred during Orthogonalization phase, Player Indexing phase, and Communication phase. Note that there is no collision in the Adaptive Learning phase. Therefore, the total collisions of DLC are given by

$$C(DLC) = C_{RP} + C_{PI} + C_{CP}. \tag{8}$$

Substitute the number of collisions in different phases from Lemmas 1, 2, and 4 in (8) we get the stated bound. We next bound the regret of DLC.

Theorem 2. *Let the conditions in Theorem 1 hold. Then with probability at least $1 - 2\delta$, the cumulative regret of* DLC *is bounded by*

$$\mathcal{R}(DLC) \leq NT(DLC).$$

Proof. As $\mu_i \in [0, 1]$ for arm $i \in [K]$, the maximum regret each player can have for any round is 1. The maximum regret for any round is bounded by N. Hence the cumulative regret of DLC is upper bounded by $NT(DLC)$. □

Corollary 1. *Let the conditions in Theorem 1 hold. The regret of DLC is bound with probability $1 - 2\delta$ is bounded as*

$$Regret(DLC) = \mathcal{O}(NK \log(K/\delta)/\Delta^2)$$

where $\Delta = \mu_N - \mu_{N+1}$.

Note that the minimizing of the sample complexity leads to lower regret for DLC. Further, the regret of DLC is linearly depended on the number of players and number of arms.

C. *Regret of* DLC vs MC

In the DLC, the number of rounds to complete all phases except Adaptive Learning phase are independent of the gap between the expected reward of the N^{th} best arm and the $(N + 1)$ best arm. AdaptiveExplore sub-routine of DLC is adapted from KL-LUCB algorithm that has the best-known performance guarantees for identifying the best N arms.

The regret of state-of-the-art MC algorithm [4][Thm. 1] is bounded by $O\left(N^2 K \log(K^2/2\delta)/\epsilon_n^2\right)$, with probability at least $1 - 2\delta$ where ϵ_n is the lower bound on the gap between the mean reward of the N^{th} best arm and the $N + 1$ best arm, i.e., $\epsilon_n < \Delta$. The value of ϵ_n is assumed to be known. The regret bound of DLC improves the bound of MC by a factor N and it is agnostic to problem-specific information like lower bound (ϵ_n). Its regret bound DLC depends on the mean reward of the arms of the given problem instance. Thus performance gain of the DLC is compared to MC is significant. As we will see later in the experimental section, the performance of DLC is an order of magnitude better than the MC algorithm. This improved performance gain is mainly due to the long random hopping phase of MC where users are made to collide for the large number of rounds which is used to estimate the number of users in the network.

IV. An Improved Adaptive Algorithm

In DLC algorithm, when the Leader explores the arms, all other players transmit only periodically and hence do not get any reward for other rounds until they are assigned an arm by the Leader. However, it was necessary for players other than the Leader to be idle so that the Leader can explore the arms freely in the Adaptive Learning phase of DLC, leading to under-utilization of arms and poor performance in term of regret. We next discuss a modification of DLC where the non-Leaders do not need to be idle while the Leader explores top N arms, which allows efficient utilization of arms resulting in the improved regret performance.

A. DLC *with Information Exchange* (DLC-Ix) *Algorithm*

In this section, we give a variant of DLC called DLC with Information Exchange (DLC-Ix) which allows the non-Leader to play their reserved arms while the Leader explores the unreserved arms and share with her the information they have gathered about their reserved arms, thus the Leader does not need to explore the reserved arms as in DLC. The DLC-Ix consists of 3 phases: 1) Orthogonalization 2) Player Indexing and 3) Adaptive Learning with Communication (ALC).

The first two phases of DLC-Ix are the same as in DLC. Once Leader is selected, other players continue to play their reserved arms until the Leader tells her to change the arm by sending signals. The Leader partitions the arms into two sets before entering to Adaptive Learning with Communication phase, namely *Reserved* and *Unreserved* arms. The set of the reserved arm of all the players except the Leader forms the Reserved set and the other set of unoccupied (empty) arms together with the reserved arm of the Leader forms the Unreserved set.

Now the Leader explores only the Unreserved set to find the best arm using the AdaptiveExplore sub-routine with $K-N+1$ arms. Once AdaptiveExplore sub-routine terminates, Leader has the best arm among Unreserved set of arms. The Leader then collides on the arm that is the reserved arm for the player with rank r. When the player observes a collision, it acts as a signal to transfer collected reward information of her reserved arm to Leader. The rewards are real-valued in $[0, 1]$ and communicating them using with arbitrary precision requires binary codes of large lengths, we thus allow the players to exchange average reward within a known fixed error. Specifically, we set the error to be ϵ_p, i.e., $(R_{T_{c,r}} - \hat{R}_{T_{c,r}})/T_{c,r} \leq \epsilon_p$ where $T_{c,r}$ is the time taken by a player with rank r to observe a collision after the end of the Player Indexing phase, R_c is the total reward collected by the player before observing collision and $\hat{R}_c$ is the total reward received by the Leader. The received reward $\hat{R}_c$ is always less than (or equal to) the collected reward R_c as some fractional part may get truncated due to the finite fixed

precision. The number of rounds needed to transfer average reward information to Leader within ϵ_p error is $\lceil \log_2(1/\epsilon_p) \rceil$[1]. In our experiments, we have set $\epsilon_p = \epsilon/2$.

DLC-Ix **DLC** with **I**nformation **Ex**change

Input: $K,\ \epsilon \geq 0,\ \delta \in (0, 1/2]$
Select the arms uniformly at random for T_{RP} rounds and settle on arm i
Play arm i for $2i$ rounds. After that sequentially hop for $K - i$ rounds and then play arm i for $K - i - 1$ rounds
Find number of players (N) and their reserved arms. Designate as Leader if has smallest reserved arm's index
if Player is Leader **then**
  for $r = 1, 2, \ldots, N-1$ **do**
    Find the best arm using *AdaptiveExplore* sub-routine in the Unreserved set of arms
    Signal player with rank r. Collect rewards information of her reserved arm and assign the current best arm to her
    Updates the Reserved and Unreserved set of arms
  end for
  Find the best arm using *AdaptiveExplore* sub-routine in the Unreserved set of arms
  Play the best arm in subsequent rounds
else
  Play arm i. When a collision is observed, send collected rewards information to Leader and after that received a arm to play in next $\lceil \log_2(K) \rceil$ rounds
  Play the received arm in subsequent rounds
end if

After receiving the mean reward from the player with rank r, Leader recomputed the best arm. If the best arm is same as the reserved arm of player, then Leader will not play it for next $\lceil \log_2 K \rceil$ rounds. Otherwise, she collides with the player for next $\lceil \log_2 K \rceil$ rounds as per the binary coding of arm index the player should take. In the former case, the player will continue to play her reserved arm in subsequent rounds. In the latter case, the player moves to the new arm assigned to her. The Leader then removes the just assigned arm from the Unreserved set and moves it to the Reserved set. The arm that is left unoccupied by this shifting is moved to the Unreserved set. The Leader then recomputes the best arm using AdaptiveExplore sub-routine and repeats the process with the next ranked player. This process stops when Leader has allocated arms to all non-Leaders from top N arms. Finally, the Leader finds the best arm among Unreserved set for herself by using AdaptiveExplore sub-routine. At the end of this process, each player has been assigned an arm from top N arms and remaining $K - N$ arms are left in the Unreserved set.

B. Analysis of DLC-Ix

We need a variant of Lemma 3 to prove sample complexity, regret and collision bounds of the DLC-Ix.

Lemma 5. *Assume technical conditions stated in Theorem 1 hold. Let* $\epsilon_p = \epsilon/2$, $c_a := (\mu_a + \mu_{a+1})/2$, *and* $m = \arg\min_{a \in [K-1]} [\min\{d(\mu_a, c_a),\ d(\mu_{a+1}, c_a)\}]$. *For any* $\epsilon \geq 0$ *and* $\delta \in (0, 1/2]$, *the ALC phase of DLC-Ix terminates with* (ϵ, δ)*-optimal allocation after* T_{ALC} *rounds with probability at least* $1 - \delta$ *where*

$$T_{ALC} \leq C_0(\alpha) H_{\epsilon, c_m} \log\left(\frac{k_1 K (H_{\epsilon,c_m})^\alpha}{\delta}\right) + (N-1)\left(\lceil \log_2(1/\epsilon) \rceil + \lceil \log_2(K) \rceil\right)$$

where $\alpha > 1, C_0(\alpha)$ *is a problem independent constant satisfying* $C_0(\alpha) \geq \alpha \log(C_0(\alpha)) + 1 + \alpha/e, k_1 > 2e + 1 + \frac{e}{\alpha - 1} + \frac{e+1}{(\alpha-1)^2}$, *and* $H_{\epsilon, c_m} := \frac{K}{\max\{d(\mu_m, c_m), \epsilon^2/2\}}$.

Proof. In case of identifying top N arms by AdaptiveExplore sub-routine, the problem dependent variable $H_{\epsilon,c}$ is defined as

$$H_{\epsilon,c} = \sum_{a \in [K]} \frac{1}{\max\{d(\mu_a, c), \epsilon^2/2\}}.$$

Since $\forall a \in [K]$, $d(\mu_m, c_m) \leq d(\mu_a, c)$,

$$H_{\epsilon,c} \leq \frac{K}{\max\{d(\mu_m, c_m), \epsilon^2/2\}} = H_{\epsilon, c_m}. \tag{9}$$

Adaptive Learning with Communication phase of DLC-Ix identifies top N arms one after other from a subset of arms (Unreserved set). It is similar to solving best arm identification problem by N number of times. By allowing information exchange between Leader and non-Leaders, this problem becomes equivalent to solve top N arms identification problem except the problem dependent variable $H_{\epsilon,c}$ can change for the given subset of arms. Therefore, we consider worst problem dependent variable H_{ϵ,c_m} that depends on the smallest gap between the mean reward of any two arms instead of the N^{th} best arm and $N+1$ best arm.

The maximum rounds needed for transferring collected mean reward of arms by non-Leaders to Leader are $\sum_{r=1}^{N-1} \lceil \log_2(2/\epsilon) \rceil$. Further, Leader also needs $\lceil \log_2(K) \rceil$ rounds to assign a arm to a non-Leader. Combine these facts with (9), we get above stated bound. □

Theorem 3. *Assume technical conditions stated in Lemma 5 hold. For any* $\epsilon \geq 0$ *and* $\delta \in (0, 1/2]$, *the sample complexity of* DLC-Ix *with probability at least* $1 - 2\delta$ *is bounded as*

$$T(\textit{DLC-Ix}) \leq \left\lceil \frac{\log(\delta/K)}{\log\left(1 - \frac{1}{4K}\right)} \right\rceil + 2K - 1 + C_0(\alpha) H_{\epsilon, c_m} \log\left(\frac{k_1 K (H_{\epsilon,c_m})^\alpha}{\delta}\right) + (N-1)\left(\lceil \log_2(2/\epsilon) \rceil + \lceil \log_2(K) \rceil\right).$$

Further with probability at least $1 - 2\delta$, *the regret of* DLC-Ix *is bounded as*

$$\mathcal{R}(\textit{DLC-Ix}) \leq N T(\textit{DLC-Ix}).$$

Proof. The total number of rounds Algorithm DLC-Ix takes for assigning (ϵ, δ)-optimal allocation to the N players is

$$T(\textit{DLC-Ix}) = T_{RP} + T_{IP} + T_{ALC}. \tag{10}$$

Substitute the number of rounds needed for different phases from Lemmas 1, 2 and 5 in (10) to get above stated bound of $T(\textit{DLC-Ix})$. As the maximum regret for any round is

[1]Note that for binary rewards $\{0, 1\}$ case, any player can transfer exact collected reward information to the Leader in the $\lceil \log_2 T_{c,r} \rceil$ rounds where the value of ϵ_p for player with rank r is $\epsilon_p(r) := 1/T_{c,r}$.

N, cumulative regret of the DLC-Ix is upper bounded by $NT(\textit{DLC-Ix})$. □

Theorem 4. *Let conditions of Lemma 5 hold. Then with probability at least* $1-2\delta$*, the maximum number of collisions that can occur in* DLC-Ix *is* $C(\textit{DLC-Ix})$ *where*

$$C(\textit{DLC-Ix}) \leq N\left\lceil \frac{\log(\delta/K)}{\log\left(1-\frac{1}{4K}\right)}\right\rceil + \frac{N(N-1)}{2} + (N-1)\left(\lceil \log_2(2/e\epsilon)\rceil + \lceil \log_2(K)\rceil\right).$$

Proof. The total collisions of DLC-Ix is is given by

$$C(\textit{DLC-Ix}) = C_{RP} + C_{PI} + C_{ALC} \tag{11}$$

The maximum number of collisions observed in Adaptive Learning with Communication phase of DLC-Ix is upper bounded by C_{ALC} where

$$C_{ALC} \leq (N-1)\left(\lceil \log_2(2/\epsilon)\rceil + \lceil \log_2(K)\rceil\right)$$

Add the number of collisions occurs in other phases from Lemmas 1 and 2 with C_{ALC} in (11) to get stated bound. □

Theorem 5. *Let* T_P *be the rounds for which non-Leader waits to play her reserved arm in Adaptive Learning phase of DLC and* R_L *is the set of reserved arms of non-Leaders after the end of the Player Indexing phase. If* T_L *is the rounds spent for exploration of top* N *arms by both DLC and DLC-Ix then*

$$\mathcal{R}(DLC)_{T_L} - \mathcal{R}(\textit{DLC-Ix})_{T_L} \geq \left(T_L - \left\lceil \frac{T_L}{T_P}\right\rceil\right)\sum_{n\in R_L}\mu_n$$

Proof. It follows from the fact that DLC needs non-Leaders play periodically in the learning phase so that the Leader can explore the arms freely whereas DLC-Ix allows non-Leaders continue to play their reserved arms in the learning phase. □

V. Experiment

We implement DLC and DLC-Ix and compare their empirical performances with state-of-the-art MC algorithm [4] and SIC-MMAB algorithm [12]. MC algorithm runs for a fixed number of rounds(T_0) and uses collisions information to find top N arms whereas SIC-MMAB has problem dependent termination rule and allows communication among players as in DLC and DLC-Ix. We repeated the experiment 20 times and plotted the cumulative regret with 95% confidence interval (the vertical line on each curve shows the confidence interval).

Figure 1 compares regret for MC, DLC and DLC-Ix using the same set of parameters used to evaluate performance of MC in [4] – $K=10, N=6, T=10000$, with mean rewards of the arms varying between 0.95 and 0.5 with separation of 0.05 between two consecutive arms. For this problem instance, MC uses $T_0 = 3000$ rounds for the learning phase, which translates the lower bound on the gap between the means as $\epsilon' = \sqrt{16K\ln(4K^2/\delta)/3000}$. We set $\epsilon(\approx \epsilon'/2) = 0.3$ and $\delta = 0.05$ in DLC and DLC-Ix for fair comparison. As expected, our algorithms have lower cumulative regret than MC.

Figure 2 compares cumulative regret of SIC-MMAB with DLC and DLC-Ix using the parameters– $K=9, N=6, T=300000, \delta=0.05$ as in [12] but we used a larger gap between the mean reward of arms for faster convergence. The mean reward of the arms varies between 0.95 and 0.55, with the same gap of 0.05 between two consecutive arms. We set $\epsilon = 0.05$ for DLC and DLC-Ix. After termination, the cumulative regret of DLC and DLC-Ix perform significantly better than SIC-MMAB. DLC does not allow non-Leader to play arm in every round during learning phase which leads to more regret in initial rounds. Whereas DLC-Ix allows non-Leader to play arm during learning phase which leads to lower regret but more sample needed for learning as shown in Figure 5.

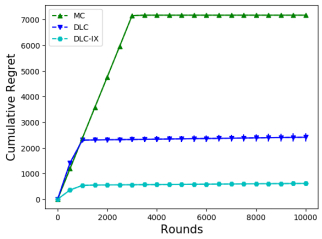

Fig. 1: Comparison with MC

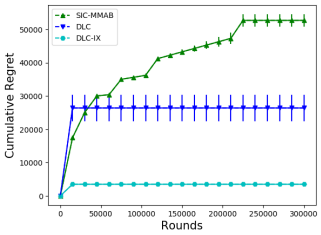

Fig. 2: Comparison with SIC-MMAB

Figures 3 and 4 depicts comparison of cumulative regret for different number of players and number of arms for a problem instance where $K=10$ (when varying number of players), $N=5$ (when varying number of arms), $T=300000, \delta=0.05, \epsilon=0.05$ and with highest mean reward set to 0.95 with others decreasing uniformly with gap of 0.05.

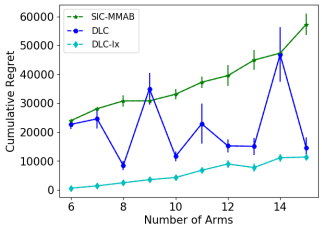

Fig. 3: Varying number of arms

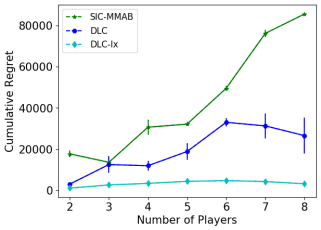

Fig. 4: Varying number of players

The cumulative regret increases with an increase in the number of arms since each arm has to be sufficiently sampled to get better estimates of mean reward. As allocations are guaranteed to be only $(\epsilon,\delta)-$optimal, sometimes DLC does not terminate with top N arms due to the bad estimate of the mean reward in the initial rounds. However, it does not happen in DLC-Ix as non-Leaders keep playing their reserved arms during exploration phase and share reward information with the Leader which helps the Leader to have good estimates for the arms she did not explore. When DLC does not terminate with top N arms, it incurs constant regret for subsequent rounds that is the reason for random behavior of DLC. Figure 3 shows that with increasing K the improvement in performance of DLC-Ix over SIC-MMAB is significant.

As the number of players increases, our algorithms perform better than SIC-MMAB as information exchange between the Leader and non-Leader happens once, whereas in SIC-MMAB it happens multiple times. SIC-MMAB runs in multiple phases where exploration and exploitation occur in each phase. At the end of exploration in each phase, players communicate the reward information of the sampled arms with other players adding a significant amount of communication overhead.

This overhead increases with increase in the number of players and arms, resulting in more regret, more rounds needed for termination and more collisions in SIC-MMAB as shown in Figure 4 and 5. The number rounds needed for the termination of DLC and DLC-Ix depend upon the gap between the mean reward of the arm and the average of the mean reward of N^{th} best arm and $N+1$ best arm (as shown in Lemma 3).

Therefore, initially regret increases with an increase in the number of players, but decreases after the number of players become more than $K/2$ as shown in Figure 4.

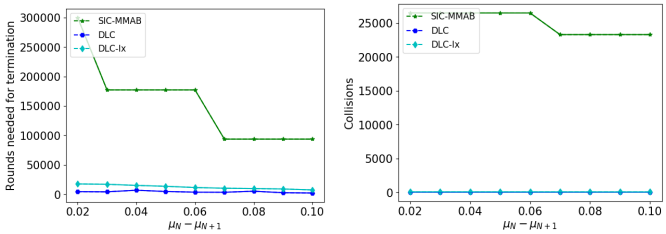

Fig. 5: Collisions and Rounds needed for termination v/s gap between consecutive arms.

DLC and DLC-Ix also have fewer collisions and need less number of rounds for finding and allocating the top N arms as compare to SIC-MMAB which is shown in Figure 5. The collisions in MC are much higher than DLC and DLC-Ix. Hence, the comparison of collisions in MC with DLC and DLC-Ix is not presented in this paper.

VI. Conclusion

We considered a CRN where multiple players access the same set of arms in absence of any central coordination. The players aim to maximize network throughput in a distributed fashion. The mean rewards of channels and number of users present in the network are unknown to the users. We set up the problem as a stochastic multi-player multi-armed bandits and developed completely decentralized and communication-free algorithms that achieve constant regret with a high probability.

In this work, we considered that the quality of the arms is the same across all the players (symmetric) and the number of users was fixed throughout (static). In the future, we would like to study the setting where the quality of arms could potentially differ across players (asymmetric) and allow the number of users to vary with time (dynamic networks).

VII. Appendix

Proof of Lemma 1: The proof of the first part of this lemma is similar to that of [18, Lemma 1], so we skip the details.

As at least two players need to play the same arm for any collision to happen, the maximum number of collisions are bounded by N for any round. Therefore, total collisions in *Orthogonalization* phase are trivially bounded by NT_{RH}. ■

Proof of Lemma 2: As the player with channel index one starts indexing other players, it needs to check the $K-1$ channel for other players and $N-1$ collisions happen during this process. The player with arm's index j starts sequential hopping after $2j$ rounds and needs to check only $K-j$ arms. She observes $N-j$ collisions in the process. The player with arm index $K-1$ starts sequential hopping after $2(K-1)$ round and need to check only the arm with index K. Therefore, the total number of rounds (T_{IP}) required to complete the *Player Indexing* phase is: $T_{IP} = 2K-1$.

The total number of collisions (C_{IP}) in the *Player Indexing* phase is bounded as: $C_{IP} = \sum_{j=1}^{N}(N-j) = N(N-1)/2$. ■

Proof of Lemma 4: Let $T_P = (N-1)(\lceil \log_2 K \rceil + 1)$ be the fixed number of rounds after which each player will play his reserved arm. In the worst case, if AdaptiveExplore sub-routine terminates just after any player played his reserved arm, then Leader has to play T_P more rounds in which he transfer arm information to other players and then takes $\lceil \log_2 K \rceil + 1$ rounds to transfer arm information to that player. Therefore, $T_{CP} \leq N(\lceil \log_2 K \rceil + 1)$.

Since Leader has to play reserved arm in the specific round before transferring the arm information to any player, this will lead to $N-1$ collisions. The arm information is transferred in its binary form which requires only $\lceil \log_2 K \rceil$ rounds. Therefore, the maximum number of collisions needed for transferring the arm information to other players is $(N-1)\lceil \log_2 K \rceil$. Therefore, the total number of collisions (C_{CP}) can occur in the *Communication* phase is bounded as: $C_{CP} \leq (N-1)(\lceil \log_2 K \rceil + 1)$. ■

Acknowledgments

Arun Verma is partially supported by MHRD Fellowship, Govt. of India. Manjesh K. Hanawal would like to thank support from INSPIRE faculty fellowships from DST, Government of India, SEED grant (16IRCCSG010) from IIT Bombay, and Early Carrier Research (ECR) Award from SERB.

References

[1] M. Ozger, F. Alagoz, and O. B. Akan, "Clustering in multi-channel cognitive radio ad hoc and sensor networks," *IEEE Communication Magazine*, vol. 56, no. 4, pp. 156–162, 2018.

[2] A. A. et al, "Channel clustering and qos level identification scheme for multi-channel cognitive radio networks," *IEEE Communication Magazine*, vol. 56, no. 4, pp. 164–171, 2018.

[3] S. Parkvall, E. Dahlman, A. Furuskár, and M. Frenne, "Nr: The new 5g radio access technology," *IEEE Communication Standards Magazine*, vol. 1, no. 4, pp. 24–30, 2017.

[4] J. Rosenski, O. Shami, and L. Szlak, "Multi-player bandits – a musical chairs approach," in *Proceedings of International Conference on Machine Learning (ICML)*, New York, USA, 2016.

[5] A. Anandkumar, N. Michael, A. K. Tang, and A. Swami, "Distributed algorithms for learning and cognitive medium access with logarithmic regret," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 731–745, 2011.

[6] O. Avner and S. Mannor, "Concurrent bandits and cognitive radio networks," in *Proceedings of the Machine Learning and Knowledge Discovery in Databases*. Stanford, CA: Springer, 2014.

[7] D. Kalathil, N. Nayyar, and R. Jain, "Decentralized learning for multiplayer multiarmed bandits," *IEEE Transactions on Information Theory*, vol. 60, no. 4, pp. 2331–2345, 2014.

[8] K. Liu and Q. Zhao, "Distributed learning in multi-armed bandit with multiple players," *IEEE Transactions on Signal Processing*, vol. 58, no. 11, 2010.

[9] L. Besson and E. Kaufmann, "Multi-player bandits revisited," in *Algorithmic Learning Theory*, 2018, pp. 56–92.

[10] M. Zandi, M. Dong, and A. Grami, "Distributed stochastic learning and adaptation to primary traffic for dynamic spectrum access," *IEEE Transactions on Wireless Communications*, vol. 15, no. 3, 2016.

[11] L. Lai, H. E. Gamal, H. Jiang, and H. V. Poor, "Cognitive medium access: Exploration, exploitation, and competition," *IEEE Transaction on Mobile Computing*, vol. 10, no. 2, 2011.

[12] E. Boursier and V. Perchet, "SIC-MMAB: Synchronisation involves communication in multiplayer multi-armed bandits," *arXiv preprint arXiv:1809.08151*, 2018.

[13] H. Tibrewal, S. Patchala, M. K. Hanawal, and S. J. Darak, "Distributed learning and optimal assignment in multiplayer heterogeneous networks," in *(to appear) IEEE International Conference on Computer Communications (INFOCOM)*, 2019.

[14] S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone, "Pac subset selection in stochastic multi-armed bandits." in *ICML*, vol. 12, 2012, pp. 655–662.

[15] E. Kaufmann and S. Kalyanakrishnan, "Information complexity in bandit subset selection," in *Conference on Learning Theory*, 2013, pp. 228–251.

[16] M. K. Hanawal and S. J. Darak, "Multi-player bandits: A trekking approach," *arXiv preprint arXiv:1809.06040*, 2018.

[17] K. Jamieson, M. Malloy, and R. Nowak, "lil'ucb:an optimal exploration algorithm for multi-armed bandits," in *Conference on Learning Theory (COLT)*, 2013.

[18] R. Kumar, A. Yadav, S. J. Darak, and M. K. Hanawal, "Trekking based distributed algorithm for opportunistic spectrum access in infrastructureless network," in *16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2018.