Learning Multiuser Channel Allocations in Cognitive Radio Networks: A Combinatorial Multi-Armed Bandit Formulation

Yi Gai, Bhaskar Krishnamachari and Rahul Jain
Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089, USA
Email: {ygai, bkrishna, rahul.jain}@usc.edu

***Abstract*—We consider the following fundamental problem in the context of channelized dynamic spectrum access. There are M secondary users and $N \geq M$ orthogonal channels. Each secondary user requires a single channel for operation that does not conflict with the channels assigned to the other users. Due to geographic dispersion, each secondary user can potentially see different primary user occupancy behavior on each channel. Time is divided into discrete decision rounds. The throughput obtainable from spectrum opportunities on each user-channel combination over a decision period is modeled as an arbitrarily-distributed random variable with bounded support but unknown mean, i.i.d. over time. The objective is to search for an allocation of channels for all users that maximizes the expected sum throughput. We formulate this problem as a combinatorial multi-armed bandit (MAB), in which each arm corresponds to a matching of the users to channels. Unlike most prior work on multi-armed bandits, this combinatorial formulation results in dependent arms. Moreover, the number of arms grows super-exponentially as the permutation $P(N, M)$. We present a novel matching-learning algorithm with polynomial storage and polynomial computation per decision period for this problem, and prove that it results in a regret (the gap between the expected sum-throughput obtained by a genie-aided perfect allocation and that obtained by this algorithm) that is uniformly upper-bounded for all time n by a function that grows as $O(M^4 N logn)$, i.e. polynomial in the number of unknown parameters and logarithmic in time. We also discuss how our results provide a non-trivial generalization of known theoretical results on multi-armed bandits.**

I. Introduction

There is considerable ongoing interest in developing dynamic spectrum access mechanisms that enable more efficient spectrum utilization [1]. Cognitive radio networks, characterized by greater levels of autonomy, intelligence and learning, are expected to play a significant role in this domain.

In this paper, we focus on a problem of fundamental significance to overlay opportunistic spectrum access. There is a set of M coordinated secondary users each trying to access one of N channels. Different from most prior work in this setting, we do not assume that each secondary user sees the same primary behavior on a given channel. This is of particular concern in geographically dispersed networks where different secondary users may be in proximity of different primary users. Thus the opportunities available on each channel might be potentially quite different for each user.

The mathematical framework we adopt in this paper is quite general. We model the throughput obtainable from exploiting the opportunities available on each user-channel combination over a decision period to be an i.i.d. random process with any arbitrary, bounded-support distribution, independent across user-channel combinations. Further it is assumed that the mean reward for each user-channel pairing is unknown to the user(it is to be determined online by a learning process). The desired objective is to maximize the expected sum-throughput of all users. Assuming an interference model whereby at most one secondary user can derive benefit from any channel, if the number of channels is greater than the number of users, the optimal channel allocation employs a one-to-one matching of users to channels, such that the expected sum-throughput is maximized.

This kind of problem, where the desired goal is to develop a sequential policy to make a selection among multiple choices, each offering stochastic rewards derived from a distribution with an unknown parameter, is traditionally formulated as an infinite horizon non-Bayesian multi-armed bandit (see [2]–[5]). A key metric of interest in evaluating a given policy for this problem is *regret*, which is defined as the difference between the expected reward gained by a genie that always makes the optimal choice, and that obtained by the given policy. The regret achieved by a policy can be evaluated in terms of its growth over time and its scalability with respect to the number of unknown parameters. Most of the prior literature on multi-armed bandits focuses on independent arms, and essentially shows logarithmic scaling of the regret over time and linear scaling with respect to the number of arms.

In this paper, we formulate the problem of learning the optimal matching of users to channels in a stochastic setting as a new kind of bandit problem that we refer to as the combinatorial multi-armed bandit. In this formulation, we map each matching (consisting of a set of component one-to-one user-channel pairings) to an arm. The reward of each arm consists of the sum of the underlying component rewards. Different from most of the classic literature, the combinatorial nature of the arms results in dependencies between them. Further, the number of arms grows super-exponentially as $P(N, M)$, the number of permutations that arrange M out of N choices.

We show first that a naive policy based on a direct application of the work by Auer *et al.* [5], that essentially ignores the dependencies between the arms, results in both storage and regret growing linearly with the large number of arms.

This raises the question whether a more sophisticated approach that exploits the combinatorial dependence between the arms can do better. We show that indeed this is possible. In particular, we develop a novel policy that we refer to as matching learning with polynomial storage (MLPS) that uses only polynomial storage and computation time at each decision period. A key sub-routine of the MLPS policy involves solving a combinatorial optimization problem pertaining to weighted matchings with polynomial complexity at each step. Our analysis of the expected regret of MLPS shows that it is bounded (for any finite n, not only asymptotically) by a function that grows much more reasonably as $O(M^4 N \log n)$ where n is the number of time steps. Thus, it is in not only logarithmic in time but also polynomial in the number of unknown parameters.

Our policy and the analysis of its regret directly generalizes the results in Auer *et al.* [5], which can be seen as the special case when $M = 1$. Moreover, though we focus on matchings because of the assumption of complete interference among the secondary users, our policy and its analysis also extends known results on MAB with multiple-plays, by allowing for arbitrary combinatorial restrictions on the set of arms played simultaneously.

Our paper is organized as follows. We first provide a survey of related work in section II. We then give a formal description of the problem we solve in section III. We first present an algorithm in section IV that is the direct application of a policy described in [5], that completely ignores the combinatorial dependencies between the arms, and show that it results in unacceptably high complexity as well as poor performance in terms of the regret as the number of users and channels is increased. We then present our MLPS policy in section V, and show that it requires only polynomial storage and polynomial computation per time period. We present the novel analysis of the regret of this algorithm in section VI and point out how this analysis generalizes known results on MAB. We discuss some of the issues pertaining to developing a protocol implementation of the proposed MLPS algorithm in cognitive radio networks for dynamic spectrum access. Finally, we conclude with a summary of our contributions and point out avenues for future work in section VIII.

II. Related Work

Lai and Robbins [2] wrote one of the earliest papers on the classic non-Bayesian infinite horizon multi-armed bandit problem. Assuming K independent arms, each generating rewards that are i.i.d. over time from a given family of distributions with an unknown real-valued parameter, they presented a general policy that provides expected regret that is $O(K \log n)$, i.e. linear in the number of arms and asymptotically logarithmic in n. They also show that this policy is order optimal in that no policy can do better than $\Omega(K \log n)$. Anantharam *et al.* [3] extend this work to the case when M simultaneous plays are allowed. The work by Agrawal [4] presents easier to compute policies based on the sample mean that also has asymptotically logarithmic regret. However, their policies cannot be directly applied to our problem formulation in this paper, which involves combinatorial arms that cannot be characterized by a single parameter.

Our work is influenced by the paper by Auer *et al.* [5] that considers arms with non-negative rewards that are i.i.d. over time with an arbitrary un-parameterized distribution that has the only restriction that it have a finite support. Further they provide a simple policy (referred to as UCB1), which achieves logarithmic regret uniformly over time, rather than only asymptotically. However, their work does not exploit potential dependencies between the arms. As we show in this paper, a direct application of their UCB1 policy therefore performs poorly for our problem formulation. Our basic policy and its analysis can be seen as a generalization of the results in [5] that allows for M plays with arbitrary constraints on the set of arms that are allowed to be played simultaneously.

While these above key papers and many others have focused on independent arms, there have been some works treating dependencies between arms. The paper by Pandey *et al.* [6] divides arms into clusters of dependent arms (in our case there would be only one such cluster consisting of all the arms). Their model assumes that each arm provide only binary rewards, and in any case, they do not present any theoretical analysis on the expected regret. In [7], the reward from each arm is modeled as the sum of a linear combination of a set of static random numbers and a zero-mean random variable that is i.i.d. over time and independent across arms. This is different from the combinatorial arm model in our paper, in which the rewards from each arm can be expressed as a linear combination of a set of independent random variables that are each i.i.d over time.

Lai *et al.* ([8], [9]) have applied multi-arm bandit formulations to user-channel selection problems in cognitive radio networks. In [8], for the case of a single user, they apply the UCB1 algorithm from [5], and for the case of decentralized multiple users they propose a randomized access policy to be applied after learning the unknown parameters. The extension of that work in [9] considers Markovian rewards and for the case of multiple users proposes a constant-probability arm-selection policy. More recently, Liu and Zhao [10] formulated the problem of secondary users selecting channels as a decentralized multi-armed bandit problem, and present a policy that achieves asymptotically logarithmic regret with respect to time. However, all these prior works applying multi-armed bandits to cognitive radio networks do not allow for the possibility that the reward process on the same channel can be different for different users. This is the key sense in which our combinatorial bandit formulation is novel. However, these works do suggest that a important direction to extend our work in the future is to consider the case of decentralized secondary users.

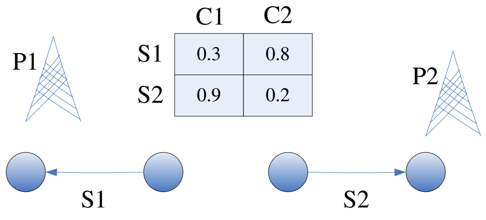
A different line of work in the domain of cognitive radio opportunistic spectrum access that has received a lot of attention recently considers dynamic decisions by a single secondary user when the underlying primary user behavior on each channel is a two-state Markov chain. This can be formulated as a POMDP, and when the channels are independent, as a special class of POMDP known as restless bandits [11]–[15]. A series of these recent results show that a surprisingly simple myopic policy is optimal when the channels are identical ([12], [13], [15]), and that this policy is the special case of Whittle's index policy for restless bandits which can be computed for non-identical channels as well [14]. Learning mechanisms for coordinating multiple users in this more complex setting are discussed in [16], [17].

There are also some other papers in the area of cognitive radio networks that focus on static user-channel matchings when the mean rewards for each combination are known *a priori*, under various assumptions about the number of users and channels, the selfishness of users, and interference [18]–[20]. Due to the emphasis of these static parameters, known distributions, however they are not directly related to multi-armed bandits.

III. Problem Formulation

There are M secondary users, and $N \geq M$ orthogonal channels. Time is divided into discrete decision periods and is denoted by the index n. At each decision period (also referred to interchangeably as time slot), each of the secondary users select a channel to sense and access according to some policy. If a secondary user i is on channel j, assuming there are no other conflicting secondary users on that channel, that user is able to opportunistically access that channel when the primary user is not occupying it, to get a non-negative stochastic throughput (reward) of $S_{i,j}(n)$. We assume that $S_{i,j}(n)$ evolves as some i.i.d. random process over time, with the only restriction that its distribution have a finite support. Without loss of generality, we normalize $S_{i,j}(n) \in [0,1]$. We do not require that $S_{i,j}(n)$ be independent across users and channels. This random process is assumed to have a mean $\theta_{i,j}$ that is unknown to the users. We denote the set of all these means as $\Theta = \{\theta_{i,j}\}$.

Figure 1 illustrates a simple scenario. There are two secondary users (i.e., links) S1 and S2, that are each assumed to be in interference range of each other. S1 is proximate to primary user P1 who is operating on channel 1. S2 is proximate to primary user P2 who is operating on channel 2. The matrix shows the corresponding Θ, i.e., the throughput each secondary user could derive from being on the corresponding channel. In this simple example, the optimal matching is for secondary user 1 to be allocated channel 2 and user 2 to be allocated channel 1. Note, however, that, in our formulation, the users are not *a priori* aware of the matrix of mean values, and therefore must follow a sequential learning policy.



	C1	C2
S1	0.3	0.8
S2	0.9	0.2

Fig. 1. An illustrative scenario

Denote by $Y_{i,j}(n)$ the actual reward obtained by a user i on channel j at time n. If user i is the only occupant of channel j, then we assume that $Y_{i,j}(n) = S_{i,j}(n)$. Else, if there are multiple users on the channel, then we assume that, due to interference, at most one of the conflicting users j' gets reward $Y_{i,j'}(n) = S_{i,j'}(n)$, while the other users on the channel $j \neq j'$ get zero reward, i.e., $Y_{i,j}(n) = 0$. This interference model covers both the perfect collision model (in which none of the conflicting users derive any benefit) and CSMA with perfect sensing (in which exactly one of the conflicting user derives benefit from the channel).

We define the deterministic policy $\pi(n)$ at each time to be a map from the observation history $\{O_k\}_{k=1}^{n-1}$ to a vector of channels $o(n)$ to be selected at period n, where user i selects channel $o_i(n)$. Then the observation history $\{O_k\}_{k=1}^{n-1}$ in turn can be expressed as $\{o_i(k), Y_{i,o_i(k)}(k)\}_{1\leq i\leq M, 1\leq k<n}$.

Since, under the assumptions above, there are sufficient channels and putting more than one user in a channel is always worse than assigning each a different channel in terms of sum-throughput, we will restrict our attention to collision-free policies that assign all users to distinct channels, which we will refer to as a permutation or matching. There are $P(N,M)$ such permutations.

Formulating our problem as a combinatorial multi-armed bandit, we map each permutation to an arm. We can represent the arm corresponding to a permutation k ($1 \leq k \leq P(N,M)$) as the index set $\mathcal{A}_k = \{(i,j) : (i,j) \text{ is in permutation } k\}$. The stochastic reward for choosing arm k at time n is then given as $Y_k(n) = \sum_{(i,j)\in\mathcal{A}_k} Y_{i,j}(n) = \sum_{(i,j)\in\mathcal{A}_k} S_{i,j}(n)$. Note that the reward obtained from each arm is i.i.d. over time, but dependent across arms that share common components.

We are interested in designing policies for this combinatorial multi-armed bandit problem that perform well with respect to *regret*, which is defined as the difference between the expected reward that could be obtained by a genie that can pick the optimal arm at each time, and that obtained by the given policy. It can be expressed as:

$$R_n^{\pi}(\Theta) = n\theta^* - E_{\pi}[\sum_{t=1}^{n} Y_{\pi(t)}(t)], \tag{1}$$

where $\theta^* = \max_k \sum_{(i,j)\in\mathcal{A}_k} \theta_{i,j}$, the expected reward of the optimal arm, is the expected sum-weight of the maximum weight matching of users to channels with $\theta_{i,j}$ as the weight.

Intuitively, we would like the regret $R_n^{\pi}(\Theta)$ to be as small as possible. If it is sub-linear with respect to time n, the time-averaged regret will tend to zero.

IV. A Naive Approach

To begin with, we show a straightforward, relatively naive approach to solving the combinatorial multi-armed bandit problem that we have defined. This approach essentially ignores the dependencies across the different arms, storing observed information about each arm independently, and making decisions based on this information alone.

In particular, we use the UCB1 policy given by Auer *et al.* [5]. In this policy, shown in Algorithm 1, two variables are stored and updated each time an arm is played: $\hat{Y}_k$ is the average of all the observation values of arm k up to the current time slot (sample mean); n_k is the number of times that arm k has been played up to the current time slot. $\hat{Y}_k$ and n_k are both initialized to 0 and updated as follows:

$$\hat{Y}_k(n) = \begin{cases} \frac{\hat{Y}_k(n-1)n_k + \sum_{(i,j)\in\mathcal{A}_k} S_{ij}(n)}{n_k(n-1)+1} & \text{, if arm k is played} \\ \hat{Y}_k(n-1) & \text{, else} \end{cases} \tag{2}$$

$$n_k(n) = \begin{cases} n_k(n-1)+1 & \text{, if arm k is played} \\ n_k(n-1) & \text{, else} \end{cases} \tag{3}$$

Algorithm 1 Policy UCB1 from Auer *et al.* [5]

```
// Initialization
Play each arm once. Update $\hat{Y}_k$, $n_k$ accordingly;
// Main loop
while 1 do
    Play arm $k$ that maximizes $\hat{Y}_k + \sqrt{\frac{2\ln n}{n_k}}$;
    Update $\hat{Y}_k$, $n_k$ accordingly;
end while
```

Theorem 1: The expected regret under UCB1 policy, specified in Algorithm 1, is at most

$$\left[8\sum_{k:\theta_k<\theta^*}\left(\frac{\ln n}{\Delta_k}\right)\right] + \left(1+\frac{\pi^2}{3}\right)\left(\sum_{k:\theta_k<\theta^*}\Delta_k\right). \tag{4}$$

where $\Delta_k = \theta^* - \theta_k$, $\theta_k = \sum_{(i,j)\in\mathcal{A}_k} \theta_{i,j}(n)$.

Proof: See [5, Theorem 1].

■

Note that in our setting UCB1 requires storage that is linear in the number of arms. The upper-bound of regret given in Theorem 1 also grows linearly with the number of arms. Since the number of arms in this formulation grows as $P(N,M)$, both of these are highly unsatisfactory. Furthermore, we conjecture that in fact even the lower bound of regret for UCB1 in this case will not be better than $\Omega(P(N,M)logn)$. This is the lower bound on the performance of any algorithm on independent arms [2], and intuitively, UCB1, as described in Algorithm 1, cannot distinguish between arms with dependent or independent rewards.

Intuitively, UCB1 algorithm performs poorly on this problem because it ignores the underlying dependencies. This motivates us to propose a sophisticated policy which more efficiently stores observations from correlated arms and exploits the correlations to make better decisions.

V. Matching Learning with Polynomial Storage

N :	number of channels.
M :	number of users, $M \le N$.
k :	index of a parameter used for an arm, $1 \le k \le P(N,M)$.
i,j :	index of a parameter used for user i, channel j.
$*$:	index indicating that a parameter is for the optimal arm.
$n_{i,j}$:	number of times that channel j has been observed by user i up to the current time slot.
$\hat{\theta}_{i,j}$:	average (sample mean) of all the observed values of channel j by user i up to the current time slot. Note that $\mathbb{E}[\hat{\theta}_{i,j}(n)] = \theta_{i,j}$.
θ_k :	$\sum_{(i,j)\in\mathcal{A}_k} \theta_{i,j}(n)$
Δ_k:	$\theta^* - \theta_k$.
Δ_{min}:	$\min_k \Delta_k$.
Δ_{max}:	$\max_k \Delta_k$.
n_i^k:	$n_{i,j}$ such that $(i,j) \in \mathcal{A}_k$ at current time slot.
$T_k(n)$:	number of times arm k has been played by MLPS in the first n time slots.
$\hat{\hat{\theta}}_k(n)$:	$\sum_{(i,j)\in\mathcal{A}_k} \hat{\theta}_{i,j}(n)$. It is the summation of all the average observation values in arm k at time n. Note that $\mathbb{E}[\hat{\hat{\theta}}_k(n)] = \theta_k$.
$\hat{T}_k(n)$:	$\min_{(i,j)\in\mathcal{A}_k} n_{i,j}(n)$.
$\hat{\theta}^k_{i,n_i^k}$:	$\hat{\theta}_{i,j}(n)$ such that $(i,j) \in \mathcal{A}_k$ and $n_{i,j}(n) = n_i^k$.
$\hat{\hat{\theta}}_{k,n_1^k,\ldots,n_M^k}$:	$\sum_{i=1}^{M} \hat{\theta}_{k,n_1^k}$.
$\hat{T}_k(n_1^k,\ldots,n_M^k)$:	$\min_i n_i^k$.

TABLE I
Notation

Table I summarizes some notation we use in the description and analysis of our algorithm.

The key idea behind this algorithm is to store and use observations for each user-channel pair, rather than for each arm as a whole. Since the same user-channel combination can occur in different matchings, this allows exploitation of information gained from the operation of one arm to make decisions about a correlated arm.

We use two M by N matrices to store the information after we play an arm at each time slot. One is $(\hat{\theta}_{i,j})_{M\times N}$ in which $\hat{\theta}_{i,j}$ is the average (sample mean) of all the observed values of channel j by user i up to the current time slot (obtained through potentially different sets of arms over time). The other one is $(n_{i,j})_{M\times N}$ in which $n_{i,j}$ is the number of times that channel j has been observed by user i up to the current time slot.

At each time slot n, after an arm k is played, we get the observation of $S_{i,j}(n)$ for all $(i,j)\in\mathcal{A}_k$. Then $(\hat{\theta}_{i,j})_{M\times N}$ and $(n_{i,j})_{M\times N}$ (both initialized to 0 at time 0) are updated as follows:

$$\hat{\theta}_{i,j}(n)=\begin{cases}\frac{\hat{\theta}_{i,j}(n-1)n_{i,j}+S_{i,j}(n)}{n_{i,j}(n-1)+1} & ,\ \text{if } (i,j)\in\mathcal{A}_k\\ \hat{\theta}_{i,j}(n-1) & ,\ \text{else}\end{cases}\tag{5}$$

$$n_{i,j}(n)=\begin{cases}n_{i,j}(n-1)+1 & ,\ \text{if } (i,j)\in\mathcal{A}_k\\ n_{i,j}(n-1) & ,\ \text{else}\end{cases}\tag{6}$$

Note that while we indicate the time index in the above updates for notational clarity, it is not necessary to store the matrices from previous time steps while running the algorithm.

Our proposed policy, which we refer to as matching learning with polynomial storage (MLPS), is shown in Algorithm 2.

Algorithm 2 Matching Learning with Polynomial Storage (MLPS)

```
// INITIALIZATION
for $p = 1$ to $M$ do
    for $q = 1$ to $N$ do
        $n = (M-1)p + q$;
        Play any permutation $k$ such that $(p,q) \in \mathcal{A}_k$;
        Update $(\hat{\theta}_{i,j})_{M\times N}$, $(n_{i,j})_{M\times N}$ accordingly.
    end for
end for
// MAIN LOOP
while 1 do
    $n = n + 1$;
    Run algorithm 3 to play arm $k$ that maximizes
        $$\sum_{(i,j)\in\mathcal{A}_k}\hat{\theta}_{i,j}+M\sqrt{\frac{(M+1)\ln n}{\min_{(i,j)\in\mathcal{A}_k} n_{i,j}}} \quad (7)$$
    Update $(\hat{\theta}_{i,j})_{M\times N}$, $(n_{i,j})_{M\times N}$ accordingly.
end while
```

Denote

$$W_k(n)=\sum_{(i,j)\in\mathcal{A}_k}\hat{\theta}_{i,j}+M\sqrt{\frac{(M+1)\ln n}{\min\limits_{(i,j)\in\mathcal{A}_k} n_{i,j}}}.\tag{8}$$

The MLPS policy chooses to play an arm with the maximum value $W_k(n)$ at each time slot to play after the initialization period when each arm is chosen once. Note that there are $P(N,M)$ arms, so using exhaustive search to solve this maximization is prohibitively expensive. We therefore propose the subroutine presented in Algorithm 3 to solve the relevant combinatorial optimization over matchings in polynomial time.

If we only focus on the first part, i.e., $\max\limits_k \sum\limits_{(i,j)\in\mathcal{A}_k}\hat{\theta}_{i,j}$, it would be the problem of finding a maximum weight matching on a labeled bipartite graph between users and channels with weights $\theta_{i,j}$. However, note that the second term in the optimization required by the MLPS policy, $\min\limits_{(i,j)\in\mathcal{A}_k} n_{i,j}$, depends on which permutation is picked, and cannot be attached to any particular edge in the maximum weight matching problem. Therefore, something additional is called for. The subroutine presented in Algorithm 3 solves the problem by conditioning on each edge one at a time to see which one fits the second term and solving for the max-weight matching over the remaining users and channels.

Algorithm 3 Matching Optimization Subroutine

Input: M, N, $(\hat{\theta}_{i,j})_{M\times N}$, $(n_{i,j})_{M\times N}$
Output: k, an arm that maximizes W_k

```
$\widetilde{\theta}_{i,j} = \hat{\theta}_{i,j}, \forall i, j$.
for $i = 1$ to $M$ do
    for $j = 1$ to $N$ do
        // ASSUME THAT $n_{i,j}$ WILL GIVE THE MAXIMUM VALUE IN (7)
        $\forall 1 \le p \le M$, set $\widetilde{\theta}_{p,j} = 0$;
        $\forall 1 \le q \le N$, set $\widetilde{\theta}_{i,q} = 0$;
        for $i' = 1$ to $M$ do
            for $j' = 1$ to $N$ do
                // DELETE AN EDGE $i'j'$ IF $n_{i'j'} < n_{i,j}$
                if $n_{i'j'} < n_{i,j}$ then
                    Set $\widetilde{\theta}_{i',j'} = 0$;
                end if
            end for
        end for
        Solve the Maximum Weight Matching problem (e.g., using the Hungarian algorithm [21]) on the bipartite graph of users and channels with edge weights $(\widetilde{\theta}_{i,j})_{M\times N}$. Call the permutation corresponding to this maximum weight matching $k_{i,j}$. Compute $W_{k_{i,j}}$ according to (8);
    end for
end for
Let $k = \arg\max_{k_{i,j}} W_{k_{i,j}}$
```

***Theorem** 2:* Algorithm 3 computes a solution to the optimization problem:

$$\max_k \sum_{(i,j)\in\mathcal{A}_k}\hat{\theta}_{i,j}+M\sqrt{\frac{(M+1)\ln n}{\min_{(i,j)\in\mathcal{A}_k} n_{i,j}}}\tag{9}$$

in polynomial time.

Proof of Theorem 2: Denote $\mathcal{O}^* = \{k : k = \arg\max\limits_k \sum\limits_{(i,j)\in\mathcal{A}_k}\hat{\theta}_{i,j}+M\sqrt{\frac{(M+1)\ln n}{\min_{(i,j)\in\mathcal{A}_k} n_{i,j}}}\}$. Denote $n_{\min}^{k^*} = \min\limits_{(i,j)\in\mathcal{A}_{k^*}} n_{i,j}^{k^*}$. We prove Theorem 2 by showing that the one of the arms in $\mathcal{O}^*$ will be obtained after running the algorithm. Specifically, we prove the following lemma.

***Lemma** 1:* $\forall k^*\in\mathcal{O}^*$, when $n_{\min}^{k^*}$ is assumed to be the one given the maximum value in (7) by the algorithm, k^* will stay as a permutation in $(\hat{\theta}'_{i,j})_{M\times N}$ for running the maximum weight matching.

Proof: When $n_{\min}^{k^*}$ is assumed, note that $\forall(i,j) \in \mathcal{A}_{k^*}, n_{i,j}^{k^*} \geq n_{\min}^{k^*}$, so none of $\hat{\theta}_{i,j:(i,j)\in\mathcal{A}_{k^*}}$ are deleted in the comparison at line 9 in Algorithm 3. Also note that k is a permutation, so $\forall(i,j) \in \mathcal{A}_{k^*}, i \neq i'$ implies $j \neq j', j \neq j'$ also implies $i \neq i'$. So none of $\hat{\theta}_{i,j:(i,j)\in\mathcal{A}_{k^*}}$ are deleted at line 5 or 6. Therefore, Lemma 1 holds.

Lemma 2: $\exists k^* \in \mathcal{O}^*$ such that k^* is the arm gotten by running the maximum weight matching at line 15.

Proof: Suppose none of the arms in $\mathcal{O}^*$ is the arm obtained by running the maximum weight matching algorithm. For any $k^* \in \mathcal{O}^*$, when $n_{\min}^{k^*}$ is assumed, there exists another arm $k' \notin \mathcal{O}^*$ such that

$$\sum_{(i,j)\in\mathcal{A}_{k'}} \hat{\theta}_{i,j} \geq \sum_{(i,j)\in\mathcal{A}_{k^*}} \hat{\theta}_{i,j}.$$

Also note that

$$\min_{(i,j)\in\mathcal{A}_{k'}} n_{i,j} = \min_{(i,j)\in\mathcal{A}_{k'}} n_{i,j} = n_{\min}^{k^*},$$

we have

$$\begin{aligned}\sum_{(i,j)\in\mathcal{A}_{k'}} \hat{\theta}_{i,j} + M\sqrt{\frac{(M+1)\ln n}{\min\limits_{(i,j)\in\mathcal{A}_{k'}} n_{i,j}}} \geq \sum_{(i,j)\in\mathcal{A}_{k^*}} \hat{\theta}_{i,j} \\ +M\sqrt{\frac{(M+1)\ln n}{\min\limits_{(i,j)\in\mathcal{A}_{k}} n_{i,j}}}.\end{aligned} \tag{10}$$

(10) implies that $k' \in \mathcal{O}^*$, which contradicts that $k' \notin \mathcal{O}^*$, therefore Lemma 2 holds.

Lemma 2 implies that k^* is one of the $k_{i,j}$ at line 18. Further, computation time of maximum weight matching on a bipartite graph is polynomial (e.g., the Hungarian method requires $O((M+N)^3)$ computations), and there are exactly $M \times N$ such computations in Algorithm 3. Hence Theorem 2 holds.

■

VI. Analysis of regret

Traditionally, the regret of a policy for a multi-armed bandit problem is upper-bounded by analyzing the expected number of times that each non-optimal arm is played, and the summing this expectation over all non-optimal arms. While such an approach will work to analyze the MLPS policy too, it turns out that the upper-bound for regret consequently obtained is quite loose, being linear in the number of arms, $P(N,M)$. Instead, we give here a tighter analysis of the MLPS algorithm that provides an upper bound which is instead polynomial in M and N and logarithmic in time. Like the regret analysis in [5], this upper-bound is also valid for finite n.

Theorem 3: The expected regret under the MLPS policy specified in algorithm 2 is at most

$$\left[\frac{4M^2(M+1)(MN)\ln n}{(\Delta_{\min})^2} + M^2N(1+\frac{\pi^2}{3})\right]\Delta_{\max}. \tag{11}$$

Proof of Theorem 3: Denote C_{t,n_j} as $M\sqrt{\frac{(M+1)\ln t}{n_j}}$. We introduce $\widetilde{T}_{ij}(n)$ as a counter after the initialization period. It is updated in the following way:

At each time slot after the initialization period, one of the two cases must happen: (1) an optimal arm is played; (2) a non-optimal arm is played. In the first case, $(\widetilde{T}_{ij}(n))_{M\times N}$ won't be updated. When an non-optimal arm $k(n)$ is picked at time n, there must be at least one $(i,j) \in \mathcal{A}_k$ such that $n_{i,j}(n) = \min\limits_{(i,j)\in\mathcal{A}_k} n_{i,j}$. If there is only one such arm, $\widetilde{T}_{ij}(n)$ is increased by 1. If there are multiple such arms, we arbitrarily pick one, say (i',j'), and increment $\widetilde{T}_{i'j'}$ by 1.

Each time when a non-optimal arm is picked, exactly one element in $(\widetilde{T}_{ij}(n))_{M\times N}$ is incremented by 1. This implies that the total number that we have played the non-optimal arms is equal to the summation of all counters in $(\widetilde{T}_{ij}(n))_{M\times N}$. Therefore, we have:

$$\sum_{k:\theta_k<\theta*} \mathbb{E}[T_k(n)] = \sum_{i=1}^{M}\sum_{j=1}^{N} \mathbb{E}[\widetilde{T}_{ij}(n)] \tag{12}$$

Also note for $\widetilde{T}_{ij}(n)$, the following inequality holds:

$$\widetilde{T}_{ij}(n) \leq n_{i,j}(n), \forall 1 \leq i \leq M, 1 \leq j \leq N. \tag{13}$$

Denote by $\widetilde{I}_{ij}(n)$ the indicator function which is equal to 1 if $\widetilde{T}_{ij}(n)$ is added by one at time n. Let l be an arbitrary positive integer. Then:

$$\begin{aligned}\widetilde{T}_{ij}(n) &= \sum_{t=MN+1}^{n} \{\widetilde{I}_{ij}(t)\} \\ &\leq l + \sum_{t=MN+1}^{n} \{\widetilde{I}_{ij}(t), \widetilde{T}_{ij}(t-1) \geq l\}\end{aligned}$$

When $\widetilde{I}_{ij}(t) = 1$, there exists some arm such that a non-optimal arm is picked for which $n_{i,j}$ is the minimum in this arm. We denote this arm as $k(t)$ since at each time that $\widetilde{I}_{ij}(t) = 1$, we could get different arms. Then,

$$\begin{aligned}\widetilde{T}_{ij}(n) &\leq l + \sum_{t=MN+1}^{n} \{\hat{\overline{\theta^*}}(t-1) + C_{t-1,\hat{T}^*(t-1)} \\ &\leq \hat{\overline{\theta}}_{k(t-1)}(t-1) + C_{t-1,\hat{T}_{k(t-1)}(t-1)}, \widetilde{T}_{ij}(t-1) \geq l\} \\ &= l + \sum_{t=MN}^{n} \{\hat{\overline{\theta^*}}(t) + C_{t,\hat{T}^*(t)} \\ &\leq \hat{\overline{\theta}}_{k(t)}(t) + C_{t,\hat{T}_{k(t)}(t)}, \widetilde{T}_{ij}(t) \geq l\}\end{aligned}$$

Note that $l \leq \widetilde{T}_{ij}(t)$ implies,

$$l \leq \widetilde{T}_{ij}(t) \leq n_{i,j}(t) = n_i^{k(t)} = \min_j n_j^{k(t)}. \tag{14}$$

This means:

$$\forall 1 \leq i \leq M, n_i^{k(t)} \geq l. \tag{15}$$

Then we have,

$$\begin{aligned}\widetilde{T}_{ij}(n) \leq\, & l + \sum_{t=MN}^{n} \{ \min_{0<n_1^*,\dots,n_M^*<t} \hat{\overline{\theta^*}}_{n_1^*,\dots,n_M^*} \\ & +C_{t-1,\hat{T}^*(n_1^*,\dots,n_M^*)} \leq \max_{l\leq n_1^{k(t)},\dots,n_M^{k(t)}<t} \hat{\bar{\theta}}_{k(t),n_1^{k(t)},\dots,n_M^{k(t)}} \\ & +C_{t-1,\hat{T}_k(n_1^{k(t)},\dots,n_M^{k(t)})} \} \\ \leq\, & l + \sum_{t=1}^{n} [\sum_{n_1^*=1}^{t-1} \cdots \sum_{n_M^*=1}^{t-1} \sum_{n_1^{k(t)}=l}^{t-1} \cdots \sum_{n_M^{k(t)}=l}^{t-1} (\hat{\overline{\theta^*}}_{n_1^*,\dots,n_M^*} \\ & +C_{t-1,\hat{T}^*(n_1^*,\dots,n_M^*)} \leq \hat{\bar{\theta}}_{k(t),n_1^{k(t)},\dots,n_M^{k(t)}} \\ & +C_{t-1,\hat{T}_{k(t)}(n_1^{k(t)},\dots,n_M^{k(t)})})]\end{aligned}$$

$\hat{\overline{\theta^*}}_{n_1^*,\dots,n_M^*} + C_{t-1,\hat{T}^*(n_1^*,\dots,n_M^*)} \leq \hat{\bar{\theta}}_{k(t),n_1^{k(t)},\dots,n_M^{k(t)}} + C_{t-1,\hat{T}_{k(t)}(n_1^{k(t)},\dots,n_M^{k(t)})}$ means that at least one of the following must be true:

$$\hat{\overline{\theta^*}}_{n_1^*,\dots,n_M^*} \leq \theta^* - C_{t-1,\hat{T}^*(n_1^*,\dots,n_M^*)} \tag{16}$$

$$\hat{\bar{\theta}}_{k(t),n_1^{k(t)},\dots,n_M^{k(t)}} \geq \theta_k + C_{t-1,\hat{T}_{k(t)}(n_1^{k(t)},\dots,n_M^{k(t)})} \tag{17}$$

$$\theta^* < \theta_k + 2C_{t-1,\hat{T}_{k(t)}(n_1^{k(t)},\dots,n_M^{k(t)})} \tag{18}$$

Now we find the upper bound for $Pr\{\hat{\overline{\theta^*}}_{n_1^*,\dots,n_M^*} \leq \theta^* - C_{t-1,\hat{T}^*(n_1^*,\dots,n_M^*)}\}$.

Note that

$$\hat{T}^*(n_1^*,\dots,n_M^*) = \min_i n_i^* \triangleq n_{\text{min}}^*,$$

We also define $n_{\text{min}}^k = \min_i n_i^k$, so $\hat{T}^k(n_1^k,\dots,n_M^k) = n_{\text{min}}^k$.

We have:

$$\begin{aligned}& Pr\{\hat{\overline{\theta^*}}_{n_1^*,\dots,n_M^*} \leq \theta^* - C_{t-1,\hat{T}^*(n_1^*,\dots,n_M^*)}\} \\ & = Pr\{\hat{\theta}_{1,n_1^*}^* + \hat{\theta}_{2,n_2^*}^* + \ldots + \hat{\theta}_{M,n_M^*}^* \leq \theta_1^* + \theta_2^* + \ldots \theta_M^* \\ & \qquad\qquad -C_{t-1,n_{\text{min}}^*}\} \\ & \leq Pr\{\text{At least one of the following must hold:} \\ & \qquad \hat{\theta}_{1,n_1^*}^* \leq \theta_1^* - \tfrac{1}{M}C_{t-1,n_{\text{min}}^*}, \\ & \qquad \hat{\theta}_{2,n_2^*}^* \leq \theta_2^* - \tfrac{1}{M}C_{t-1,n_{\text{min}}^*}, \\ & \qquad\qquad \vdots \\ & \qquad \hat{\theta}_{1,n_M^*}^* \leq \theta_M^* - \tfrac{1}{M}C_{t-1,n_{\text{min}}^*}\} \\ & \leq Pr\{\hat{\theta}_{1,n_1^*}^* \leq \theta_1^* - \tfrac{1}{M}C_{t-1,n_{\text{min}}^*}\} \\ & \quad +Pr\{\hat{\theta}_{2,n_2^*}^* \leq \theta_2^* - \tfrac{1}{M}C_{t-1,n_{\text{min}}^*}\} + \ldots \\ & \quad +Pr\{\hat{\theta}_{M,n_M^*}^* \leq \theta_M^* - \tfrac{1}{M}C_{t-1,n_{\text{min}}^*}\}.\end{aligned}$$

$\forall 1 \leq i \leq M$, applying the Chernoff-Hoeffding bound [24], we could find the upper bound of each item in the above equation as,

$$\begin{aligned}& Pr\{\hat{\theta}_{i,n_i^*} \leq \theta_i^* - \tfrac{1}{M}C_{t-1,n_{\text{min}}^*}\} \\ & = Pr\{n_i^*\hat{\theta}_{i,n_i^*} \leq n_i^*\theta_i^* - \tfrac{n_i^*}{M}C_{t-1,n_{\text{min}}^*}\} \\ & \leq e^{-2\cdot\frac{1}{n_i^*}\cdot(n_i^*)^2\cdot\frac{(M+1)\ln t}{n_{\text{min}}^*}} \\ & \leq e^{-2\cdot\frac{1}{n_i^*}\cdot(n_i^*)^2\cdot\frac{(M+1)\ln t}{n_i^*}} \\ & = e^{-2(M+1)\ln t} \\ & = t^{-2(M+1)}.\end{aligned}$$

Thus,

$$Pr\{\hat{\overline{\theta^*}}_{n_1^*,\dots,n_M^*} \leq \theta^* - C_{t-1,\hat{T}^*(n_1^*,\dots,n_M^*)}\} \leq Mt^{-2(M+1)}. \tag{19}$$

Similarly, we can get the upper bound of the probability for inequality (17):

$$\begin{aligned}& Pr\{\hat{\bar{\theta}}_{k(t),n_1^{k(t)},\dots,n_M^{k(t)}} \geq \theta_k + C_{t-1,\hat{T}_{k(t)}(n_1^{k(t)},\dots,n_M^{k(t)})}\} \\ & \quad \leq Mt^{-2(M+1)}.\end{aligned} \tag{20}$$

Note that for $l \geq \left\lceil \frac{4(M+1)\ln n}{\left(\frac{\Delta_{k(t)}}{M}\right)^2} \right\rceil$,

$$\begin{aligned}& \theta^* - \theta_{k(t)} - 2C_{t-1,\hat{T}_{k(t)}(n_1^{k(t)},\dots,n_M^{k(t)})} \\ & = \theta^* - \theta_{k(t)} - 2M\sqrt{\frac{(M+1)\ln t}{n_{\text{min}}^{k(t)}}} \\ & \geq \theta^* - \theta_{k(t)} - M\sqrt{\frac{4(M+1)\ln n}{n_{\text{min}}^{k(t)}}} \\ & \geq \theta^* - \theta_{k(t)} - M\sqrt{\frac{4(M+1)\ln n}{4(M+1)\ln n}\left(\frac{\Delta_{k(t)}}{M}\right)^2} \\ & = \theta^* - \theta_{k(t)} - \Delta_{k(t)} = 0.\end{aligned} \tag{21}$$

(21) implies that condition (16) is false when $l = \left\lceil \frac{4(M+1)\ln n}{\left(\frac{\Delta_{k(t)}}{M}\right)^2} \right\rceil$. If we let $l = \left\lceil \frac{4(M+1)\ln n}{\left(\frac{\Delta_{\text{min}}^{i,j}}{M}\right)^2} \right\rceil$, then (16) is false for all $k(t), 1 \leq t \leq \infty$ where

$$\Delta_{\text{min}}^{i,j} = \min_k\{\Delta_k : (i,j) \in \mathcal{A}_k\}. \tag{22}$$

Therefore,

$$\begin{aligned}\mathbb{E}[\widetilde{T}_{ij}(n)] \leq\, & \frac{4(M+1)\ln n}{\left(\frac{\Delta_{\text{min}}^{i,j}}{M}\right)^2} \\ & + \sum_{t=1}^{\infty}\left(\sum_{n_1^*=1}^{t-1} \cdots \sum_{n_1^*=M}^{t-1} \sum_{n_1^k=1}^{t-1} \cdots \sum_{n_1^k=M}^{t-1} 2Mt^{-2(M+1)}\right) \\ \leq\, & \frac{4M^2(M+1)\ln n}{\left(\Delta_{\text{min}}^{i,j}\right)^2} + M\sum_{t=1}^{\infty} 2t^{-2} \\ \leq\, & \frac{4M^2(M+1)\ln n}{\left(\Delta_{\text{min}}^{i,j}\right)^2} + M(1 + \tfrac{\pi^2}{3})\end{aligned} \tag{23}$$

So under our MLPS policy,

$$
\begin{aligned}
R_n^\pi(\Theta) &= \theta^* n - \mathbb{E}_\pi[\sum_{t=1}^{n} Y_{\pi(t)}(t)] \\
&= \sum_{k:\theta_k<\theta*} \Delta_k \mathbb{E}[T_k(n)] \\
&\le \Delta_{\max} \sum_{k:\theta_k<\theta*} \mathbb{E}[T_k(n)] \\
&= \Delta_{\max} \sum_{i=1}^{M} \sum_{j=1}^{N} \mathbb{E}[\widetilde{T}_{ij}(n)] \\
&\le \left[\sum_{i=1}^{M} \sum_{j=1}^{N} \frac{4M^2(M+1)\ln n}{(\Delta_{\min}^{i,j})^2} + M^2 N(1+\frac{\pi^2}{3}) \right] \Delta_{\max} \\
&\le \left[\frac{4M^2(M+1)(MN)\ln n}{(\Delta_{\min})^2} + M^2 N(1+\frac{\pi^2}{3}) \right] \Delta_{\max}
\end{aligned}
\tag{24}
$$

■

We note that when $M = 1$, which means there is only one user in the system, the upper bound of regret in Theorem 3 becomes the same as the upper bound in Theorem 1 in [5]. So Theorem 3 is a more general result.

Remark: Because of the application context of cognitive radio networks, we have focused on policies that search only through matchings in this paper. But we note that the MLPS policy we have described and its analysis can be extended easily to a more general combinatorial multi-armed bandit problem in which arms represent any arbitrary collection (of size no more than M) of components. In this case, the subroutine in Algorithm 3 will no longer apply as it is specific to finding a matching to optimize the relevant quantity. But essentially everything else about the MLPS policy in Algorithm 2 (which should perhaps be then renamed "combinatorial-arm learning with polynomial storage" or CLPS, to deemphasize matchings) can remain the same.

VII. Implementation Considerations

The MLPS algorithm we have presented can be run either at a centralized coordinator or in a distributed manner by each secondary user (link). We discuss briefly some of the pertinent issues in translating the algorithm to a practical protocol.

For an example of the first setting, consider a scenario where the secondary users are nodes communicating with a common access point. The access point can then be in charge of announcing (over some predetermined control channel) the non-conflicting channels to be used by each user for each decision period after running the MLPS algorithm. In an uplink setting, the throughput from each secondary user on the corresponding channel can be measured directly by the access point. For downlink settings, the throughput may be measured by the access point through acknowledgement or direct signaling from the secondary user.

Even if there is no common access-point for all secondary users, it is possible to develop a protocol involving some form of distributed leader election [22] to appoint a coordinator to which information about the measured throughput on each secondary user is communicated, and which runs the MLPS algorithm. Given the assumption in this paper that all secondary users are in interference range of each other, it is not entirely unreasonable to assume that they can communicate to a common node.

If there are compelling reasons to implement this protocol in a distributed manner (e.g., to avoid having a central point of failure), an alternative approach is to design a protocol whereby the users of the secondary network can reliably propagate their channel measurements to all other secondary users once each decision period, and have each secondary user run the MLPS algorithm independently over the common information. This is similar to how distributed link state routing protocols such as OLSR [23] propagate link state information and compute routes. In this case, care must be taken to have deterministic tie-breaking where needed, to maintain consistency between the secondary users (for instance if there are multiple permutations that are solutions to the optimization subroutine).

A further level of decentralization that may be desired in some settings is for the users to not propagate any information explicitly, and instead rely on purely local observations and decisions. Such a decentralized solution has been developed recently for the conventional multi-armed bandit setting where all users perceive the same rewards for each channel [10]. However, developing such a completely decentralized version of the matching learning policy for the problem of combinatorial multi-armed bandits described in this paper remains an open question.

Our description of the MLPS algorithm also leaves open the question of the granularity of each decision period. The optimal duration may have to be determined empirically taking into account constraints such as the minimum time needed to accommodate all the communication and computation required at each step.

VIII. Conclusion

We have presented in this paper a new kind of bandit problem that we refer to as a combinatorial multi-armed bandit. The key distinction of this formulation from the classic non-Bayesian multi-armed bandit problem is each arm is itself a combinatorial "bundle" of components. The number of arms is consequently quite large and there are dependencies between the rewards provided by arms sharing common components. A slightly different formulation of our problem is to think of arms as being the components themselves, with multiple plays allowed simultaneously. But from this perspective too, there is a key difference from prior work: in our formulation, there are pre-specified restrictions on exactly which bundles of plays are allowed simultaneously.

We have shown that a naive approach that ignores the dependencies between the combinatorial arms requires too much storage and also performs poorly as the regret scales linearly with the number of arms (while we have rigorously proved only the scaling of the upper bound, we conjecture that this is true for the lower bound as well in this case).

We have therefore developed a more sophisticated policy that we refer to as matching learning with polynomial storage (MLPS). This policy stores only information about the component user-channel pairings and uses a polynomial time matching optimization at each time step. Our analysis of this policy is unique in that we bound the number of times the large number of non-optimal arms are visited by tracking only a polynomial number of quantities. This results in an upper bound on the expected regret that scales only polynomially in the number of components. As we noted, while we have focused on permutations/matchings in this paper, the basic algorithm and analysis are in fact directly generalizable to arbitrary restrictions on the set of sub-components that be selected at each time. Our algorithm and analysis are therefore a direct generalization of the results provided by Auer *et al.* [5] for non-combinatorial arms.

The theoretical formulation we have presented in this paper is of fundamental relevance to dynamic spectrum access using cognitive radio networks. When there are geographically dispersed secondary users it is reasonable to assume that they experience different primary user behavior on each channel. While we have focused on the simplest complete-interference case when all secondary users interfere with each other and must therefore each be allocated distinct channels, it is easy to conceive of a generalization of this work where the interference constraints are specified on a graph, with a formulation that takes into account coloring constraints. We plan to explore this direction further in the future.

As noted before, an interesting open problem is to develop a completely decentralized version of this policy in which secondary users do not have to share any information with each other and act purely on local observations. Another open question at present is to derive a lower bound on the regret that can be achieved by any policy for the combinatorial multi-armed bandit. Since there are $M \times N$ independent components that must be explored, we conjecture that this lower bound is $\Omega(MN \log n)$.

Acknowledgment

We would like to thank David Kempe for his valuable input on the matching optimization subroutine. We would also like to thank Maheswaran Sathiamoorthy and Harsha Honnappa for useful discussions.

References

[1] Q. Zhao and B. Sadler, “A survey of dynamic spectrum access: signal processing, networking, and regulatory policy,” *IEEE Signal Processing Magazine*, vol. 55, no. 5, 2007.

[2] T. Lai and H. Robbins, “Asymptotically efficient adaptive allocation rules,” *Advances in Applied Mathematics*, vol. 6, no. 1, 1985.

[3] V. Anantharam, P. Varaiya, and J. Walrand, “Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part I: IID rewards,” *IEEE Tran. on Auto. Control*, vol. 32, no. 11, 1987.

[4] R. Agrawal, “Sample mean based index policies with $O(\log n)$ regret for the multi-armed bandit problem,” *Advances in Applied Probability*, 27.

[5] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multiarmed bandit problem,” *Machine Learning*, 47(2-3), 2002.

[6] S. Pandey, D. Chakrabarti, and D. Agarwal, “Multi-armed bandit problems with dependent arms,” In *24th Intl. Conf. on Machine Learning*, pp. 721-728, 2007.

[7] P. Rusmevichientong and J. N. Tsitsiklis, “Linearly parameterized bandits,” CoRR, abs/0812.3465, 2008.

[8] L. Lai, H. E. Gamal, H. Jiang, and H. V. Poor, “Cognitive medium access: exploration, exploitation and competition,” [Online]. Available: http://arxiv.org/abs/0710.1385.

[9] L. Lai, H. Jiang and H. Vincent Poor, “Medium access in cognitive radio networks: a competitive multi-armed bandit framework,” in *Proc. of IEEE Asilomar Conference on Signals, Systems, and Computers*, CA, Oct. 2008.

[10] K. Liu and Q. Zhao, “Decentralized multi-armed bandit with multiple distributed players,” Technical Report (TR-09-03), Sep. 2009, http://www.ece.ucdavis.edu/∼qzhao/TR-09-03.pdf.

[11] Y. Chen, Q. Zhao, and A. Swami, “Joint design and separation principle for opportunistic spectrum access in the presence of sensing errors,” *IEEE Transactions on Information Theory*, vol. 54, no. 5, May, 2008.

[12] Q. Zhao, B. Krishnamachari, and K. Liu, “On myopic sensing for multi-channel opportunistic access: structure, optimality, and performance,” *IEEE Transactions on Wireless Communications*, 2008.

[13] S. Ahmad, M. Liu, T. Javidi, Q. Zhao, and B. Krishnamachari, “Optimality of myopic sensing in multi-channel opportunistic access,” *IEEE Transactions on Information Theory*, 2009.

[14] K. Liu and Q. Zhao, “A restless bandit formulation of opportunistic access: indexablity and index policy,” *Proc. of the 5th IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops*, June, 2008.

[15] S. H. A. Ahmad and M. Liu, “Multi-channel opportunistic access: a case of restless bandits with multiple plays,” *Allerton Conference*, 2009.

[16] H. Liu, B. Krishnamachari, and Q. Zhao, “Cooperation and learning in multiuser opportunistic spectrum access,” in *Proc. of IEEE International Conference on Communications Workshops*, May, 2008.

[17] K. Liu, Q. Zhao, and Y. Chen, “Distributed sensing and access in cognitive radio networks,” in *Proc. of 10th International Symposium on Spread Spectrum Techniques and Applications* , August, 2008

[18] H. Liu and B. Krishnamachari, “Randomized strategies for multi-user multi-channel opportunity sensing,” *Cognitive Radio Networks Workshop*, IEEE CCNC 2008.

[19] H. Liu, B. Krishnamachari and Q. Zhao, “Negotiating multichannel sensing and access in cognitive radio wireless networks”, *IEEE SECON Workshop on Software Defined Radio*,June, 2009.

[20] H. Liu, A. B. MacKenzie and B. Krishnamachari, “Bargaining to improve channel sharing between selfish cognitive radios”, *IEEE Globecom*, December 2009.

[21] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, 1955.

[22] N. Lynch, *Distributed Algorithms*, 1996.

[23] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum and L. Viennot, “Optimized link state routing protocol,” *IEEE International Multipotic Conference*, 2001.

[24] D. Pollard, *Convergence of Stochastic Processes*. Berlin: Springer, 1984.