DORA: Distributed Cognitive Random Access of Unslotted Markovian Channels under Tight Collision Constraints

Liqiang Zhang
Dept. of Computer and Information Sciences
Indiana University South Bend
South Bend, USA
Email: liqzhang@iusb.edu

***Abstract*—We consider the design of distributed strategies that allow multiple secondary users to opportunistically access multiple unslotted Markovian channels with unknown parameters and tight collision constraints, a challenging problem setting that has not been well addressed by existing work. An optimal strategy would strike a balance among exploration, which is to measure all the channels to identify the best one(s), exploitation, which is to stay on the currently best channel(s) as much as possible, and competition, that is to spread out users in order to avoid overcrowding the best channel(s). Moreover, a strategy has to abide collision constraint of each channel to become an acceptable one. We first assume known channel parameters and formulate a CNLP (constrained nonlinear programming) problem, which we solve through an algorithm we call DORA-Known that computes an optimal randomized access strategy. Next, We address the online channel-parameter learning problem by transforming it into a problem of DTMC (discrete-time Markov chain) estimation with incomplete data, and solving it with an EM (expectation-maximization) based algorithm. We then propose an algorithm called DORA-Learning that extends DORA-Known to incorporate the online channel learning. The proposed algorithms are evaluated and compared with a state-of-art approach that assumes known channel parameters, and two reinforcement learning based schemes. Experimental results illustrate significant performance gain of the two DORA algorithms over the other three approaches.**

I. Introduction

The co-existence of perceived scarcity of wireless spectrum and many significantly underutilized licensed bands has urged more efficient and dynamic spectrum allocation and use. As the key enabling technology to the so called "Dynamic Spectrum Access (DSA)", Cognitive Radio (CR) allows unlicensed users, called secondary users (SUs), to identify the "spectrum holes" of licensed channels and utilize them opportunistically as long as they do not cause collisions to the communications of licensed users, called primary users (PUs), beyond a prescribed level. In presence of multiple channels, a key decision for an SU to determine is which channel(s) to sense. Designing efficient cognitive access strategies for SUs to maximize the utilization of spectrum opportunities has been one of the most active research topics in CR.

The problem of finding optimal cognitive access strategy has been studied under different settings. When single SU is considered, the major challenge is to find the optimal tradeoff between *exploration* and *exploitation*. On one hand, the user needs to sufficiently explore all the primary channels to identify the best one(s). On the other hand, the time spent on exploring inferior channels should be minimized so that the best one(s) can be better utilized. The generalization to the case of multiple SUs brings another dimension to the problem, *competition*. The channel selection strategy must take into account that the good channel(s) may be desired by other contending SUs as well. Crowding SUs into the best channel(s) will lead to opportunities on others unexploited.

Most existing work on cognitive access has assumed slotted (i.i.d. or Markovian) primary channel models. Recently, a more realistic unslotted Markovian model [9], [10] has started to receive attention. With unslotted models, a primary user may start to transmit at any time. This leads to potential collisions with SUs in spite of perfect sensing during sensing period. Therefore, enforcing collision constraints for unslotted primary channels is a nontrivial issue even assuming no sensing errors.

In this work, we consider multiple unslotted Markovian channels with unknown parameters that are opportunistically accessed by multi-users constrained by tight collision constraints. We aim at distributed strategies that maximize the utilization of the spectrum opportunities. Despite extensive research effort in the related area, the problem under this setting has not been well studied. Existing work either focuses on single user case [15], [21]–[24], or assumes slotted primary channels [1], [5]–[8], [15], [16], [21]–[23], or assumes known channel parameters [12], [13], [23], [24].

We tackle the problem by taking a two-stage approach. First we assume channel parameters are known a priori and formulate a CNLP (constrained nonlinear programming) problem, for which we propose an distributed algorithm called DORA-Known that computes an optimal randomized access strategy. Next, We address the online channel-parameter learning problem by transforming it into a problem of DTMC (discrete-time Markov chain) estimation with incomplete data, which is solved through an EM (expectation-maximization) based algorithm. We then propose an algorithm called DORA-Learning that extends DORA-Known to incorporate the online

channel learning.

The rest of the paper is organized as follows. We first provide a brief survey of related work in Section II. We then present the system model in Section III. In Section IV, we formulate the problem, discuss the optimal solution, and present the DORA-Known algorithm. In Section V, we address the online channel-parameter estimation problem and present the DORA-Learning algorithm. Performance evaluation is given in Section VI. Finally, Section VII concludes the paper.

II. Related Work

Cognitive access with single SU has been extensively studied in the literature, where it is often modeled as a multi-armed bandit (MAB) problem [15], [21] or a more generalized form – partially observed Markov decision problem (POMDP) [22], [23].

The case for multiple SUs with slotted primary traffic model has been investigated by [1], [5]–[8], [12], [16], where the problem has been formulated as MAB with multi-players [12], [16], combinatorial-MAB [5]–[7], or distributed-MAB [8].

Unslotted Markovian channel model was first considered in [24] with the setting of single SU, where the authors formulate the problem as a constrained Markov decision process (CMDP) through applying a periodic sensing policy. One heuristic protocol introduced in [24], PS-MA (periodic sensing with memoryless access), was shown optimal under tight collision constraints by a later work [13]. The extension of PS-MA to the multi-user scenario has also been addressed in [13], where both a centralized version, called OPS-MA, and a distributed version, which we referred to as RPS-MA, were presented. OPS-MA has been shown optimal, however, it has serious limitations: it needs a central controller and requires the number of SUs to be less than the number of channels. We compare our approach with RPS-MA through simulations in VI.

A learning-based approach under the setting of multiple SUs and unslotted Markovian channels was proposed by Shetty et al. in [19], where a constrained POMDP problem is formulated. However, they assume channel parameters are known a priori and an SU can distinguish PUs' traffic from other SUs, neither is assumed in our work.

In [11], Kim and Shin formulated two optimization problems with the objectives set as maximal discovery of spectrum opportunities and minimum channel-switching latency respectively. They considered a more general unslotted semi-Markov channel model. However, their approach assumes (1) all SUs form a single -hop network, and (2) all SUs tune to the same channel all the time in a synchronous manner. None of them is required in our work.

III. System Model

A. Channel and Sensing Model

We consider N parallel primary channels, each with bandwidth B, and K secondary users that opportunistically exploit the spectrum holes of the primary channels. The usage pattern of each primary channel is modeled as a continuous time Markov ON-OFF process alternating between ON (busy) and OFF (idle) periods exponentially distributed with mean μ_i^{-1} and λ_i^{-1}, respectively. We assume the states of different channels evolve independently to one another. The state transition rate matrix for the ith channel is given by

$$Q_i \triangleq \begin{pmatrix} -\lambda_i & \lambda_i \\ \mu_i & -\mu_i \end{pmatrix}.$$

We assume non-stationary, block-varying primary channels, which means the Q-matrices stay fixed for a block of time units and randomly change at the beginning of the next block.

Each SU is assumed to be equipped with an antenna that can be tuned to any primary channel for sensing and accessing. Primary channels are *symmetric* to SUs, meaning that at any time, if any two SUs tune to the same primary channel, they would observe the same channel status if no sensing error. Similar assumption has been made in most prior work.

We assume that SUs employ time-slotted transmissions. Each slot consists of a sensing window and a transmission window. We adopt the concept of *quite period* introduced in IEEE 802.22 by making each sensing window a quite period, during which all SUs suspend their transmission so that the status of a primary channel can be sensed without interference. Note that we assume SU has no ability to discern between the transmission of an SU and that of a PU. We denote the slot size by T, and the size of sensing windows by T_s. At each sensing window, each SU makes a decision to choose a channel to sense and, in case it is idle, access, in the transmission window. Aiming at distributed solutions, we focus on randomized channel selection strategies. An strategy, which we denote as $\vec{\rho}$, can be described by a vector of possibilities:

$$\vec{\rho} = (\rho_1, ..., \rho_N),$$

where ρ_i stands for the possibility of choosing channel i to sense and access, and $\sum_{i=1}^{N} \rho_i \leq 1$. Note that each SU may dynamically vary its strategy over time.

The channel and sensing model is illustrated in Fig. 1 with an example of three primary channels and 2 SUs. As shown in the figure, an SU might collide with PUs' transmission even the channel is sensed idle earlier during the sensing window, since an PU might come back at any time. And it should also be noted that, with the setting of multiple SUs, it is possible that two or more SUs choose to sense and access the same channel. In this case, they will share the transmission window through certain contention resolution mechanism, which we will address in next subsection.

B. Basic MAC Model of Secondary Users

When multiple SUs transmit in the same primary channel within the same slot, activities of these SUs must be coordinated to avoid collisions. A simple solution is to divide each transmission window into mini-slots and equip each SU with a Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) based protocol. This approach comes with two types of overhead. First, the idle mini-slots involved in the back-off procedures cannot be utilized; and second, there is

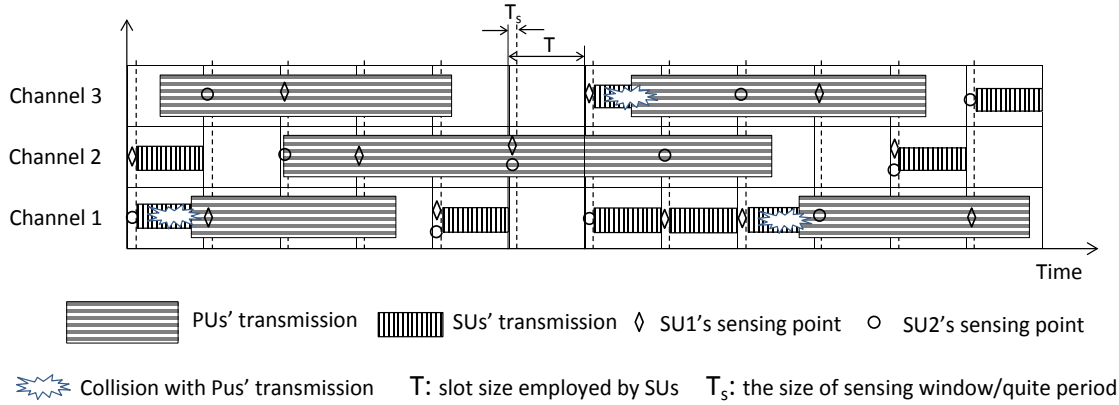


Fig. 1. The Illustration of the Channel and Sensing Model (an example of 3 unslotted primary channels cognitively accessed by 2 SUs employing slotted transmissions).

still a possibility of collisions that cannot be avoided by the protocol. The amount of the overhead, which impacts the utilization of the shared slot, in general, depends on the number of SUs sharing the slot. However, as illustrated by Bianchi in [2], if each contending station is tuned to an optimal contention window, the maximum utilization (i.e., the maximum aggregate throughput) stays around 0.82, independent of the number of the contending stations. Based on this, we assume the utilization of a shared slot, as long as no collision with PUs' transmission gets involved, remains a constant, which we denote as U, no matter how many SUs are sharing that specific slot.

It should be pointed out, however, that the collision avoidance mechanism employed by SUs does not eradicate the collisions with PUs' transmission for two reasons. Firstly, the carrier sensing employed in CSMA/CA is often too short compared to the incumbent sensing. Secondly, PUs are not expected to have similar (collision avoidance) mechanism built in. A PU might come back during the transmission of an SU, and therefore results a collision.

We assume some synchronization mechanism is in place so that the transmitting SU and its intended receiver are always tuned to the same channel at the same time. If no sensing error is assumed, this can be achieved through sharing a same-seeded random number generator. Otherwise, a common control channel (CCC) based scheme [17] can be applied, which is beyond the scope of this work.

IV. Distributed Optimal Access Strategy with Known Channel Parameters

Our objective is to find optimal strategy for each SU so that the utilization of the spectrum opportunities is maximized (or equivalently, the aggregate throughput of all SUs is maximized). For now, we assume the channel parameters (e.g., Q-matrices) are known a priori.

A. Problem Formulation

1) Reward Function: We first focus on a block of L time slots during which the parameters of primary channels stay fixed. Once we are able to compute the optimal strategies for one block, we only need to repeat the computation for all the blocks. Fixed channel parameters imply a static strategies over the block of L slots. On the other hand, since primary channels are symmetric to SUs, it implies the optimal strategies should also be symmetric to SUs. Therefore, we only need to find one single optimal strategy $\vec{\rho}^*$ that is shared by all SUs:

$$\vec{\rho}^* = \arg\max_{\vec{\rho}} R(\vec{\rho}, L), \tag{1}$$

where $R(\vec{\rho}, L)$ denote the aggregate throughput of all SUs over the L slots, and can be expressed as follows:

$$R(\vec{\rho}, L) = B(T-T_s)UL\sum_{i=1}^{N}(1-(1-\rho_i)^K)\frac{\mu_i}{\lambda_i+\mu_i}exp(-\lambda_i T),$$

where B is the channel bandwidth, $(T - T_s)$ the size of transmission window, and U the single slot utilization constant introduced in III-B. $\frac{\mu_i}{\lambda_i+\mu_i}exp(-\lambda_i T)$ represents the probability of channel i being idle in the beginning of a slot and remains idle throughout that slot, where $\frac{\mu_i}{\lambda_i+\mu_i}$ is the probability of channel i being idle in the beginning of a slot and $exp(-\lambda_i T)$ the probability it stays unchanged for a period of T. $(1-\rho_i)^K$ represents the probability that none of the K SUs chooses channel i, and therefore $(1-(1-\rho_i)^K)$ is the probability that at least one SU chooses channel i.

2) Collision Constraints: The solution to equation (1) might not be an acceptable strategy since it might lead to collisions with PUs' transmission beyond a prescribed tolerance level, which we denote by γ_i for channel i. To address this issue, we define the measure for the degree of the interference caused by SUs on a primary channel as the asymptotic ratio of collisions and the number of slots during which PUs are active. The collision constraints can then be expressed as

$$\frac{(1-(1-\rho_i)^K)\frac{\mu_i}{\lambda_i+\mu_i}(exp(-\lambda_i T_s)-exp(-\lambda_i T))}{1-\frac{\mu_i}{\lambda_i+\mu_i}exp(-\lambda_i T)} \leq \gamma_i, \tag{2}$$

where the denominator represents the probability that channel i is used by PUs, and $\frac{\mu_i}{\lambda_i+\mu_i}(exp(-\lambda_i T_s)-exp(-\lambda_i T))$ the probability that a slot remains idle throughout the sensing window but becomes busy during the transmission window.

Now, after simplification and let

$$\hat{\gamma_i} \triangleq \gamma_i \frac{1 + \frac{\lambda_i}{\mu_i} - exp(-\lambda_i T)}{exp(-\lambda_i T_s) - exp(-\lambda_i T)}, \quad (3)$$

the collision constraint represented by in-equation (2) can be written as

$$1 - (1 - \rho_i)^K \leq \hat{\gamma_i},$$

or equivalently,

$$\rho_i \leq 1 - (max(0, 1 - \hat{\gamma_i}))^{1/K}.$$

3) CNLP Formulation: Our objective is to maximize $R(\vec{\rho}, L)$. With B, $T - T_s$, U, all being constants, and L is given, $\max R(\vec{\rho}, L)$ is equivalent to

$$\min \sum_{i=1}^{N} (1 - \rho_i)^K \frac{\mu_i}{\lambda_i + \mu_i} exp(-\lambda_i T).$$

Therefore, we have the following constrained nonlinear optimization formulation of the problem:

$$\begin{aligned} \text{minimize:} & f = \sum_{i=1}^{N} (1 - \rho_i)^K \frac{\mu_i}{\lambda_i + \mu_i} exp(-\lambda_i T), \\ \text{subject to} & \\ & \rho_i \leq 1 - (max(0, 1 - \hat{\gamma_i}))^{1/K}, \forall i, \\ & \sum_{i=1}^{N} \rho_i \leq 1, \\ & \rho_i \geq 0. \end{aligned} \quad (4)$$

B. Optimal Access Strategy

To find the optimal solution for the formulated problem, we divide it into two cases, namely, when $K = 1$, and when $K > 1$.

When $\mathbf{K = 1}$, the problem is reduced to a constrained linear programming problem. It is straightforward to obtain the optimal solution that follows. Sort the channels according to $\frac{\mu_i}{\lambda_i + \mu_i} exp(-\lambda_i T)$ in decreasing order. Without loss of generality, let $i_1, i_2, ..., i_N$ denote the indices of the channels after being sorted with i_1 being the channel with highest rank. Then the optimal strategy is

$$\begin{aligned} \rho^*_{i_1} &= \min(1, \hat{\gamma_{i_1}}), \\ \rho^*_{i_2} &= \min(1 - \rho_{i_1}, \hat{\gamma_{i_2}}), \\ &\cdots \\ \rho^*_{i_N} &= \min(1 - \sum_{j=1}^{N-1} \rho_{i_j}, \hat{\gamma_{i_N}}). \end{aligned} \quad (5)$$

This is consistent to our intuition. With single SU, the "best" channel(s) should be exploited with maximal probability only limited by the collision constraints. For the extreme case, when $\hat{\gamma_{i_1}} \geq 1$, the SU will simply stick to the channel that provides best spectrum opportunity. Note, this is possible even if this channel is associated with a tight collision constraint since $\hat{\gamma_{i_1}}$ also depends on channel parameters (as shown in (3)).

When $\mathbf{K > 1}$, we have the following observations: (1) The objective function f is a continuously differentiable convex function. This is easy to verify since all the mixed partials of f are zero and $\frac{\partial^2 f}{\partial \rho_i^2} \geq 0$ for each $i = 1, \cdots, N$. (2) All the inequality constraints are also continuously differentiable convex functions. Therefore, the KKT (Karush-Kuhn-Tucker) conditions are necessary and also sufficient for optimality:

$$\begin{aligned} \omega_i + \psi - K(1 - \rho_i)^{K-1} \frac{\mu_i}{\lambda_i + \mu_i} exp(-\lambda_i T) &= 0, \\ \rho_i \leq 1 - (\max(0, 1 - \hat{\gamma_i}))^{1/K}, & \\ \omega_i(\rho_i - 1 + (\max(0, 1 - \hat{\gamma_i}))^{1/K}) &= 0, \\ \psi(\sum_{i=1}^{N} \rho_i - 1) &= 0, \\ \sum_{i=1}^{N} \rho_i \leq 1, & \\ \omega_i \geq 0, & \\ \psi \geq 0, & \\ \rho_i \geq 0, & \end{aligned}$$

where $\omega_i (i = 1, \cdots N)$, and ψ are KKT multipliers.

The optimal solution depends on the value of $\hat{\gamma_i}$s. It is easy to see that when $\sum_{i=1}^{N} (1 - (max(0, 1 - \hat{\gamma_i}))^{1/K}) < 1$, $\sum_{i=1}^{N} \rho_i$ can only be less than one. Therefore, ψ has to be zero, which leads to the conclusion that ω_i cannot be zero for any i. It follows that $\rho_i^* = 1 - (\max(0, 1 - \hat{\gamma_i}))^{1/K}, \forall i$. Noticing that the condition $\sum_{i=1}^{N} (1 - (max(0, 1 - \hat{\gamma_i}))^{1/K}) < 1$ implies $\hat{\gamma_i} < 1, \forall i$, the optimal solution can be simply written as

$$\rho_i^* = 1 - (1 - \hat{\gamma_i})^{1/K}, \forall i. \quad (6)$$

It is worthwhile noting that, in this case, with probability $1 - \sum_{i=1}^{N} \rho_i$, the SU will not sense and access any channel. This seemingly "wasted" opportunity is due to the collision constraints put on the channels to protect primary users. When channel parameters are unknown, however, a SU can utilize the opportunity to sense (but not transmit within) channels for the purpose of channel estimation. Further discussion will be presented in next section.

When $\sum_{i=1}^{N} (1 - (max(0, 1 - \hat{\gamma_i}))^{1/K}) \geq 1$, it is easy to verify that the following solution satisfy the KKT conditions:

$$\rho_i^* = \begin{cases} min(max(0, & \\ 1 - (\frac{\psi(\lambda_i + \mu_i)}{K \mu_i exp(-\lambda_i T)})^{1/(K-1)}), & \\ 1 - (max(0, 1 - \hat{\gamma_i}))^{1/K}), & \text{for } \mu_i > 0, \\ 0, & \text{for } \mu_i = 0. \end{cases} \quad (7)$$

where ψ is a constant that satisfies $\sum_{i=1}^{N} \rho_i = 1$.

It should be noted that even though all the SUs are to follow the same single optimal strategy, they need to compute this strategy individually in a distributed manner.

Require: the number of channels N, the number of SUs K, channel parameters $\vec{\lambda} = (\lambda_1, ..., \lambda_N)$, $\vec{\mu} = (\mu_1, ..., \mu_N)$, and collision constraint $\vec{\gamma} = (\gamma_1, ..., \gamma_N)$. Initialize $slotIndex$ to be 0 when joining the network.

if receive updated $[\vec{\lambda}, \vec{\mu}]$ **then**
  $\vec{\rho}$ = calculateStrategy($\vec{\lambda}$, $\vec{\mu}$, $\vec{\gamma}$)
end if
if $rand() < 1 - \sum_{i=1}^{N} \rho_i$ **then**
  not to choose any channel in next slot
else
  choose channel i with probability $\rho_i / \sum_{i=1}^{N} \rho_i$ in next slot
end if
$slotIndex \leftarrow slotIndex + 1$

function calculateStrategy($\vec{\lambda}$, $\vec{\mu}$, $\vec{\gamma}$)
calculate $\hat{\vec{\gamma}}$ = $(\hat{\gamma_1}, ..., \hat{\gamma_N})$ acc. (3)
if $K = 1$ **then**
  sort channels acc. $\frac{\mu_i}{\lambda_i + \mu_i} exp(-\lambda_i T)$ in decreasing order
  calculate strategy $\vec{\rho}$ acc. (5)
else
  if $\sum_{i=1}^{N} (1 - (max(0, 1 - \hat{\gamma_i}))^{1/K}) < 1$ **then**
    calculate strategy $\vec{\rho}$ acc. (6)
  else
    calculate strategy $\vec{\rho}$ acc. (7)
  end if
end if
return $\vec{\rho}$

Fig. 2. **Algorithm 1** (DORA-Known: distributed cognitive random access with known channel parameters)

We summarize the channel selection strategy for the case with known channel parameters in Algorithm 1 as shown in Fig. 2. The algorithm, which we refer to as DORA-Known, is to be executed by each SU before each sensing window. Note that the channel selection strategy is recalculated every time channel parameters are updated.

V. Learning Unknown Channel Parameters

When channel parameters (λ_i, μ_i) are unknown a priori, we need to estimate these parameters online based on the channel sensing results. Since channels are chosen and sensed in a randomized manner, the samplings for any specific channel are irregular in terms of sampling intervals. This poses a special challenge to the channel estimation.

Fig. 3 illustrates an example of irregular samplings on a group of three channels, where an asterisk refers to the absence of data at the particular time point. Our objective is then to estimate the transition rate matrix for each channel through those samplings. We show that this CTMC (continuous time Markov chain) estimation problem can be converted to a DTMC (discrete time Markov chain) estimation problem with incomplete data, and the latter can then be solved through an EM (expectation-maximization) based algorithm.

A. Problem Transformation

Note that each SU node i needs to individually estimate the transition rate matrix for each channel j, or it is to say that, for the SU network, there is a whole matrix of Q_j^i to be estimated dynamically over time. For ease of presentation, however, we drop the subscripts i and j for now, focusing only on one of such transition rate matrices, denoted as

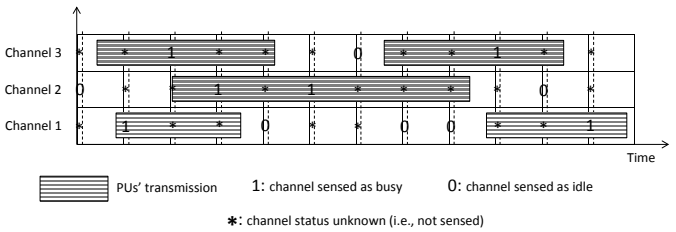


Fig. 3. The Illustration of Irregular Samplings by an SU Node.

$$Q = \begin{pmatrix} -\lambda & \lambda \\ \mu & -\mu \end{pmatrix}.$$

First, it is easy to see that, for a continuous-time Markov process $X(t)$ with transition rate matrix Q, if it is periodically sampled with period T, then the samples, $Z(k) = X(kT), k = 1, 2, ...,$ can be modeled as a discrete-time Markov process with transition matrix

$$P = exp(QT) = \begin{pmatrix} 1 - p_{01} & p_{01} \\ p_{10} & 1 - p_{10} \end{pmatrix}, \tag{8}$$

where p_{01} (p_{10}) is the transition probability from state 0 (1) to 1 (0), and

$$\begin{aligned} p_{01} &= \frac{\lambda}{\lambda + \mu}(1 - exp(-(\lambda + \mu)T)), \\ p_{10} &= \frac{\mu}{\lambda + \mu}(1 - exp(-(\lambda + \mu)T)). \end{aligned} \tag{9}$$

Obviously, as long as P is estimated, Q will be instantly available. As the following equations can be directly obtained from (9)

$$\begin{aligned} \lambda &= \frac{-p_{01} \ln(1 - p_{01} - p_{10})}{(p_{01} + p_{10})T}, \\ \mu &= \frac{-p_{10} \ln(1 - p_{01} - p_{10})}{(p_{01} + p_{10})T}. \end{aligned} \tag{10}$$

Now, if the complete series, or a continuous block, of $Z(k)$ are available, then P can be easily estimated through an MLE (maximum likelihood estimator) [3]. The challenge here, however, is that we do not have continuous samples from each channel, as a channel is chosen by an SU in a randomized manner.

B. The EM-based Algorithm

Suppose that we have a vector of r samples from a channel, $\underline{Z} = (Z_{s_1}, Z_{s_2}, \cdots, Z_{s_r})$, where $s_k (k = 1, 2, \cdots, r)$ denotes the index of slots during which the samples are made. Let $\underline{P}$ denote the transition matrix to be estimated. Applying the Markovian property, we have the following likelihood of

$$\underline{N}_{ij}(\underline{P}^{(l)}) = \sum_{m=0}^{1}\sum_{n=0}^{1}\sum_{t=1}^{S} O_{mnt} \sum_{k=0}^{t-1} \frac{((\underline{P}^{(l)})^k)_{mi}(\underline{P}^{(l)})_{ij}((\underline{P}^{(l)})^{(t-k-1)})_{jn}}{((\underline{P}^{(l)})^t)_{mn}}. \tag{14}$$

observed data $\underline{Z}$ given $\underline{P}$:

$$\begin{aligned}\mathcal{L}(\underline{P};\underline{Z}) &= P(\underline{Z} \mid \underline{P}) \\ &= Pr(Z_{s_1} = z_1;\underline{P})\cdot \\ &\quad \prod_{k=1}^{r} Pr(Z_{s_k} = z_k \mid Z_{s_{k-1}} = z_{k-1};\underline{P}) \\ &= Pr(Z_{s_1} = z_1;\underline{P}) \prod_{k=2}^{r} P_{z_{k-1}z_k}(s_k - s_{k-1});\underline{P}),\end{aligned}$$

where $P_{z_{k-1}z_k}(s_k - s_{k-1})$ denotes the probability that a sample z_{k-1} is followed by a sample z_k and the inter-sample collection time is $(s_k - s_{k-1})$ slots. Now, let S denote the biggest inter-sample span between two consecutive samples in $\underline{Z}$, O_{ijt} the number of observed transitions from state i to state j occurring over t time units (e.g., slots), and $(\underline{P}^t)_{ij}$ the ijth component of the matrix $\underline{P}^t$, then the likelihood can be re-written as:

$$\mathcal{L}(\underline{P};\underline{Z}) = Pr(Z_{s_1} = z_1;\underline{P}) \prod_{i=0}^{1}\prod_{j=0}^{1}\prod_{t=1}^{S}((\underline{P}^t)_{ij})^{O_{ijt}},$$

and the log-likelihood is as follows:

$$\ln \mathcal{L}(\underline{P};\underline{Z}) = \ln Pr(Z_{s_1} = z_1;\underline{P}) + \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{t=1}^{S} O_{ijt} \ln(\underline{P}^t)_{ij}.$$

Note that when S equals 1, the problem reduces to the transition matrix estimation of DTMC with complete data, and the likelihood function can be easily expressed in a mathematical form. However, when S is greater than 1, it is too complex to analytically maximize the likelihood function. In this case, EM-based algorithm can be applied to solve the problem.

The EM algorithm is an interactive method used to find the maximum likelihood parameters of a statistical model in cases where the model depends on unobserved variables [4], [14], [18]. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to update the expectation in the next E step.

In our case, the parameter to estimate is the transition matrix $\underline{P}$. We start the EM algorithm by assigning it an initial value $\underline{P}^{(0)}$. With this current estimate of $\underline{P}$, the E-step reconstructs a "complete"set of data, $\underline{Y}$ from the"incomplete", observed data $\underline{Z}$. In the lth iteration of the E-step, it computes the expectation of the log-likelihood of the "complete" data with $\underline{P}$, given the current estimate $\underline{P}^{(l)}$ and the "incomplete" data $\underline{Z}$:

$$Q(\underline{P} \mid \underline{P}^{(l)}) = \mathbb{E}[\ln \mathcal{L}(\underline{P};\underline{Y}) \mid \underline{Z}, \underline{P}^{(l)}]. \tag{11}$$

Require: the observed incomplete data $\underline{Z}$ (data samples and indices of slots during which the samples are made), convergence criterion ϵ.

// Randomly choose a starting transition matrix $\underline{P}^{(0)}$
$\underline{p}_{01}^{(0)} \leftarrow rand()$
$\underline{p}_{10}^{(0)} \leftarrow rand()$
$\underline{p}_{00}^{(0)} \leftarrow 1 - \underline{p}_{01}^{(0)}$
$\underline{p}_{11}^{(0)} \leftarrow 1 - \underline{p}_{10}^{(0)}$
find largest inter-sample span S from $\underline{Z}$
calculate matrix O for $\underline{Z}$ // each element of O, O_{mnt} $(m = 0, 1,\ n = 0, 1,\ t = 1, ..., S)$ represents the number of observed transitions from state m to state n occurring over t slots.
$l \leftarrow 0$
loop
  calculate $\underline{P}^{(l+1)}$ acc. (13) and (14).
  if $\max(\max(|\underline{P}^{(l+1)} - \underline{P}^{(l)}|)) < \epsilon$ **then**
    break
  end if
  $l \leftarrow l + 1$
end loop
return $\underline{P}^{(l+1)}$

Fig. 4. **Algorithm 2** (The EM-based algorithm for estimating transition matrix $\underline{P}$ based on incomplete samples $\underline{Z}$)

The M-step then computes a new estimate:

$$\underline{P}^{(l+1)} = \arg\max_{\underline{P}} Q(\underline{P} \mid \underline{P}^{(l)}). \tag{12}$$

The E and M-steps are repeated until the sequence $\{\underline{P}^{(l)}\}$ converges.

Due to space limitation, we skip the expression and analysis for equations (11) and (12), and directly give the mathematical form for $\underline{P}^{(l+1)}$ as follows:

$$(\underline{P}^{(l+1)})_{ij} = \frac{\underline{N}_{ij}(\underline{P}^{(l)})}{\sum_{k=0}^{1} \underline{N}_{ik}(\underline{P}^{(l)})}, \tag{13}$$

where $i, j = 0, 1$, and $\underline{N}_{ij}(\underline{P}^{(l)})$ is given by (14).

Our implementation of the EM algorithm, as shown in Fig. 4, is largely based on [18]. In our simulation, the algorithm always converges even we choose to start with a random transition matrix. However, as pointed out in [18], the EM algorithm might often converge to a local maximum or a saddle point, not necessarily a global maximum. To increase the possibility of finding global maximum, a simple solution is to just repeat the algorithm several times, each with a (different) random starting matrix, and pick the $\underline{P}$ yielding the highest maxima.

Now, with the estimated DTMC transition matrix $\underline{P}$ from irregular samples $\underline{Z}$, an estimate of channel parameters (λ, μ) is instantly available through equation (10).

C. Trading Exploitation for Exploration When Necessary

To effectively estimate channel parameters, there is yet another issue we need to address. An optimal access strategy may choose to visit certain channels with very low probabilities. In case this happens, the time needed to collect sufficient samples for estimating those channels could be significantly lengthened, which might lead to a slow-reaction to channel changes. Therefore, we need to balance the exploitation and exploration to guarantee that all the channels are visited at least with a certain threshold probability, for example, 1%, which we denote by ρ_{low}.

To this end, we introduce an complementary strategy vector, $\vec{\underline{\rho}} = (\underline{\rho_1}, ..., \underline{\rho_N})$. For any channel i, if $\rho_i < \rho_{low}$, it is compensated with $\underline{\rho_i} = \rho_{low} - \rho_i$. Different from ρ_i, with probability of $\underline{\rho_i}$, channel i will be sensed but not accessed, since we do not want to violate the collision constraint of the channel. It is worthwhile noting that we need to keep $\sum_{i=1}^{N}(\rho_i + \underline{\rho_i}) <= 1$. To accommodate $\vec{\underline{\rho}}$, we first exploit the "leftover" opportunity $1 - \sum_{i=1}^{N} \rho_i$ if any. In case it is not sufficient, we then squeeze opportunities from those channels with $\rho_i > \rho_{low}$ proportionally so that $\rho_i + \underline{\rho_i} >= \rho_{low}$ holds for any channel.

The complete algorithm for distributed cognitive random access with online channel-parameter learning, which we referred as DORA-Learning, is shown in Fig. 5. The algorithm bootstraps by letting each SU periodically sense, but not access, each channel during the first $sWindow*N$ slots (lines 1-3). This is to let the SU collect sufficient samples from each channel so that an initial estimate of all the channels can be made. Accessing the channels at this stage are not allowed since uniformly accessing each channel might lead to collision constraint violation. For any channel, if an SU has collected $sWindow$ new samples from it, its channel parameters need to be (re)estimated (lines 6-8). Channel selection strategy need to recalculated if the parameters for any channel has been updated (lines 10-12). As discussed earlier, besides normal sensing and accessing (line 16), DORA-Learning allows channel i to be sensed but not accessed with probability $\underline{\rho_i}$ (line 17) to ensure sufficient samples collected from each channel while not violating collision constraints. The function for calculating strategy is given in lines 21-44, where the computation outlined by lines 22-32 remains same as in Algorithm 1. Lines 33-43 describes the procedure of tuning $\vec{\rho}$ and $\vec{\underline{\rho}}$ as discussed in the last paragraph.

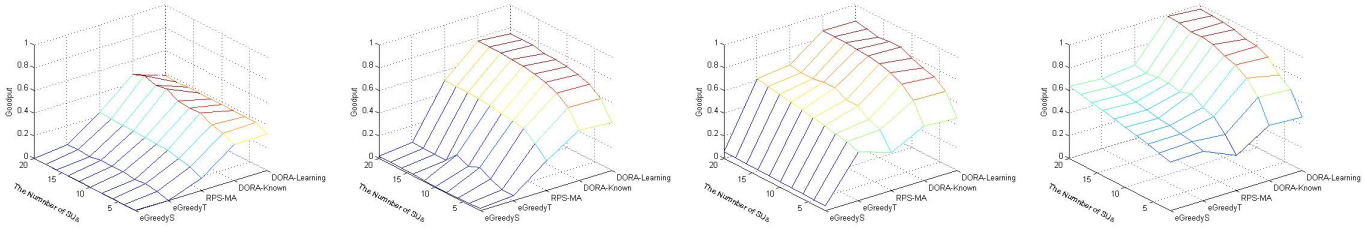
A non-zero $\underline{\rho}$ would, in most cases (e.g., the cases when the condition in line 38 is true), indicate a trade of exploitation for exploration. This trade will surely hurt the optimality of the original strategy, however, only to an minimal and controlled level (as $\underline{\rho}$ is bounded by ρ_{low}). And the trade does not always happen.

Require: the number of channels N, the number of SUs K, and collision constraints $\vec{\gamma} = (\gamma_1, ..., \gamma_N)$. Initialize $slotIndex$ to be 0 when joining the network.

// Bootstrapping (lines 1-3) : periodically sense each channel
// to get initial samples.
1: **if** $slotIndex < sWindow * N$ **then**
2: $i \leftarrow slotIndex \bmod N$
3: sense but not access channel $i + 1$
4: **else**
5: **for** $i = 1$ **to** N **do**
6: **if** $sWindow$ new samples have been collected for channel i **then**
7: estimate/update (λ_i, μ_i) acc. Algorithm 2 and (10)
8: **end if**
9: **end for**
10: **if** (λ_i, μ_i) has been updated for at least one i **then**
11: [$\vec{\rho}$, $\vec{\underline{\rho}}$] = calculateStrategy($\vec{\lambda}$, $\vec{\mu}$, $\vec{\gamma}$)
12: **end if**
13: **if** $rand() < 1 - \sum_{i=1}^{N}(\rho_i + \underline{\rho_i})$ **then**
14: randomly choose a channel to sense but not access in next slot
15: **else**
16: with probability $\rho_i / \sum_{i=1}^{N}(\rho_i + \underline{\rho_i})$, choose channel i to sense and access in next slot
17: with probability $\underline{\rho_i} / \sum_{i=1}^{N}(\rho_i + \underline{\rho_i})$, choose channel i to sense but not access in next slot
18: **end if**
19: **end if**
20: $slotIndex \leftarrow slotIndex + 1$

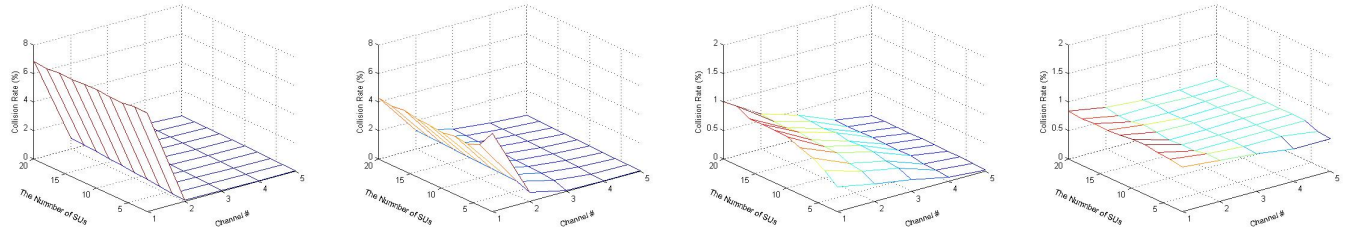
21: **function** calculateStrategy($\vec{\lambda}$, $\vec{\mu}$, $\vec{\gamma}$)
22: calculate $\vec{\hat{\gamma}} = (\hat{\gamma_1}, ..., \hat{\gamma_N})$ acc. (3)
23: **if** $K = 1$ **then**
24: sort channels acc. $\frac{\mu_i}{\lambda_i + \mu_i} exp(-\lambda_i T)$ in decreasing order
25: calculate strategy $\vec{\rho}$ acc. (5)
26: **else**
27: **if** $\sum_{i=1}^{N}(1 - (max(0, 1 - \hat{\gamma_i}))^{1/K}) < 1$ **then**
28: calculate strategy $\vec{\rho}$ acc. (6)
29: **else**
30: calculate strategy $\vec{\rho}$ acc. (7)
31: **end if**
32: **end if**
33: $\vec{\underline{\rho}} = (0, ..., 0)$
34: **repeat**
35: **for** each i with $\rho_i < \rho_{low}$ **do**
36: $\underline{\rho_i} \leftarrow \rho_{low} - \rho_i$
37: **end for**
38: **if** $\sum_{i=1}^{N} \underline{\rho_i} > 1 - \sum_{i=1}^{N} \rho_i$ **then**
39: **for** each i with $\rho_i > \rho_{low}$ **do**
40: $\rho_i \leftarrow \rho_i - (\sum_{i=1}^{N} \underline{\rho_i} - (1 - \sum_{i=1}^{N} \rho_i)) * \frac{\rho_i}{\sum_{i=1}^{N}(\rho_i * 1_{[\rho_i > \rho_{low}]})}$
// $1_{[]}$ is the indicator function.
41: **end for**
42: **end if**
43: **until** $\rho_i + \underline{\rho_i} >= \rho_{low}$ holds for each i
44: **return** [$\vec{\rho}$, $\vec{\underline{\rho}}$]

Fig. 5. **Algorithm 3** (DORA-Learning: distributed cognitive random access with online channel-parameter learning)



(a) Collision constraint: 1% (b) Collision constraint: 3% (c) Collision constraint: 5% (d) No collision constraint

Fig. 6. Goodput for different collision constraint settings



(a) eGreedyS (b) eGreedyT (c) RPS-MA (d) DORA-Learning

Fig. 7. Channel collision rates for the case with 1% collision constraint

VI. Performance Evaluation

A. Reference Schemes

To measure the performance of the proposed approaches, we conduct extensive Matlab simulations and compare them with the following reference schemes:

- **RPS-MA**. As introduced in Section II, RPS-MA is a distributed multi-user cognitive access strategy presented in [13], which we consider as the most closely related work. The scheme shares almost all the same assumptions as our work, except it assumes channel parameters are known a priori.
- **eGreedyS**. This scheme is based on the classic reinforcement learning method ϵ-greedy [20] for the MAB problem. Each SU runs the algorithm independently, in which an SU's action is defined as selecting a channel to sense, and the reward is defined based on the sensing result. If a chosen channel is sensed as "idle", the reward is 1, otherwise, the SU gets zero reward.
- **eGreedyT**. This scheme is just a different variant of eGreedyS. In this version, the reward is defined differently. If a chosen channel is sensed as "idle", the reward is not necessarily 1, instead, it is 1 divided the number of SUs that chooses this same channel at the slot. We expect that, with this definition of reward, the SUs can somehow "sense" the competition from other SUs.

B. Performance Metrics

We define two performance metrics: **goodput** and **collision rate**. The goodput is defined as the ratio of utilized spectrum opportunities and the total opportunities. The collision rate is measured for each channel as the ratio of the number of collisions recorded and the total number of slots that PUs are active. Note, that in one test, if any channel's collision constraint is violated (e.g., measured collision rate is higher than the collision constraint), then the goodput for that specific test is counted as zero. For the results presented in this section, we repeat each experiment for 10 tests and report their average.

C. Simulation Setup

Parameter	Value/Value Range
N (the number of channels)	5
K (the number of SUs)	2 to 20
Channel Parameters ($[\lambda_i^{-1}, \mu_i^{-1}]$)	[9, 1], [7, 3], [5, 5], [3, 7], [1, 9]
γ (collision constraint, set same for all channels)	1%, 3%, 5%, or 100%
T (slot size)	$0.25s$
T_s (sensing window size)	$0.01s$
$sWindow$	200
Simulation Time	$10000s$

TABLE I
The setting of parameters for the simulation

We summarize the configuration and settings for the simulation in Table I. Note the channels are configured to present different levels of spectrum opportunities. For example, the average idle and busy durations of the first channel are 9 and 1 seconds respectively, while the last channel has the reverse. In the simulation, channel parameters are only given (as input) to RPS-MA and DORA-Known, as the other three schemes assume unknown channel parameters. We have tested various channel configurations. Due to space limitation, however, only the one shown in the table is reported.

D. Numerical Results

Fig. 6 compares the goodput for different schemes under various collision constraint settings. As it is clearly shown in the figure, the two DORA algorithms achieve significant higher goodput than the other schemes, even when the collision constraints are lifted up (e.g., set as 100%). eGreedyT performs better than eGreedyS as the former can "sense" the competition therefore adjusts its policy accordingly. However, both algorithms performs poorly when the collision constraints decreases to 3% or lower, as they would violate the constraints in almost every run of the test.

The collision rate of the channels under the 1% collision constraint is shown in Fig. 7. The result for DORA-Known is similar to DORA-Learning, and therefore, is skipped to save space. As we can see from the figure, with eGreedyS, channel 1 bears almost all the collisions while leaving other channels way below the collision constraint. This is understandable since the nature of the scheme is to attract SUs to the best channel. eGreedyT does a little better job by shifting more channel accesses to other channels, however, not able to lower down the collision rate of channel 1 to an acceptable level. On the other hand, RPS-MA and the two DORA algorithms were able to balance the channel accesses in a better way so that more spectrum opportunities are discovered while at the same time abiding the collision constraint of each channel.

VII. Conclusion and Future Work

We studied distributed cognitive access strategies for multi-users that opportunistically explore multiple unslotted Markovian channels with tight collision constraints, and proposed two algorithms: DORA-Known and DORA-Learning. The former assumes known channel parameters and is an optimal solution. The latter assumes no prior knowledge about channel parameters and employs an EM-based algorithm to learn the channels online. DORA-Learning is quasi-optimal as it trades exploitation for exploration when necessary to ensure good estimation of all channels. Simulation results demonstrate that the proposed approaches are able to achieve high utilization of spectrum opportunities without violating the tight collision constraints of the channels.

When sensing error is considered, an SU has to be more conservative in order to strictly abide collision constraints. Both DORA algorithms can be extended to tolerate certain degree of sensing errors. In the current work, we assume each SU can only sense one channel at a time. The extension to multi-channel sensing (e.g., with multi-antennas) will be studied in the future. A more ambitious plan would consider more general channel models, for example, a semi-Markovian channel model.

Acknowledgment

The research research was supported in part by an Indiana University Faculty Research Grant.

References

[1] A. Anandkumar, N. Michael and A. Tang, "Opportunistic Spectrum Access with Multiple Users: Learning under Competition," *in Proc. of IEEE INFOCOM'10"*, San Diego, CA, March 2010.

[2] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, No. 3, pp. 535-547, Mar. 2000.

[3] B. A. Craig and P. P. Sendi, "Estimation of the transition matrix of a discrete-time Markov chain," *Health Economics*, vol. 11, no. 1, 2002.

[4] A. O. Dempster, N. M. Laird and D. B. Rubin, "Maximum-Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statist. Soc. Ser. B.*, vol. 39, no. 1, 1977.

[5] Y. Gai, B. Krishnamachari and R. Jain, "Learning Multiuser Channel Allocation in Cognitive Radio Networks: A Combinatorial Multi-Armed Bandit Formulation," *in Proc. of IEEE International symposium on Dynamic Spectrum Access Networks (DySPAN, 10)*, Singapore, April 2010.

[6] ——, "Combinatorial Network Optimization with Unknown Variables: Multi-Armed Bandits with Linear Rewards and Individual Observations," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, 2012.

[7] Y. Gai, B. Krishnamachari and M. Liu, "On the Combinatorial Multi-Armed Bandit Problem with Markovian Rewards," *in Proc. of IEEE Globecom'11*, Houston, TX, Dec. 2011.

[8] Y. Gai and B. Krishnamachari, "Decentralized Online Learning Algorithms for Opportunistic Spectrum Access," *in Proc. of IEEE Globecom'11*, Houston, TX, Dec. 2011.

[9] S. Geirhofer, L. Tong and B. M. Sadler, "Dynamic Spectrum Access in WLAN Channels: Empirical Model and Its Stochastic Analysis," *in Proc. of First International Workshop onTechnology and Policy for Accessing Spectrum (TAPAS'06)*, Boston, MA, August 2006.

[10] S. Geirhofer, L. Tong and B. M. Sadler, "A Measurement-Based Model for Dynamic Spectrum Access in WLAN Channels," *in Proc. of IEEE MILCOM 2006*, Washington, DC, Oct. 2006.

[11] H. Kim and K. G. Shin, "Efficient Discovery of Specturm Opportunities with MAC-Layer Sensing in Cognitive Radio Networks," *IEEE Trans. Mobile Computing*, vol. 7, no. 5. May 2008.

[12] L. Lai, H. E. Gamai, H, Jiang and V. Poor, "Cognitive Medium Access: Exploration, Exploitation, and Competition," *IEEE Trans. Mobile Computing*, vol. no 2. Feb. 2011.

[13] X. Li, Q. Zhao, X. Guan and L. Tong, "Optimal Cognitive Access of Markovian Channels under Tight Collision Constraints," *IEEE J. Sel. Areas Commun.*, vol. 29, no.4, April 2011.

[14] R. J. A. Little and D. B. Rubin, "Statistical Analysis with Missing Data," *Wiley-Interscience*, 2 edition, September 2002.

[15] H. Liu, K. Liu and Q. Zhao, "Learning in A Changing World: Restless Multi-Armed Bandit with Unknown Dynamics," *http://arxiv.org/abs/1011.4969*, v2 (last revised in Dec. 2011).

[16] K. Liu and Q. Zhao, "Distributed Learning in Multi-Armed Bandit With Multiple Players," *IEEE Trans. Signal Processing*, vol. 58, no. 11, Nov. 2010.

[17] B. F. Lo, "A survey of common control channel design in cognitive radio networks," *Physical Communication*, vol. 4, no. 1, March 2011.

[18] C. Sherlaw-Johnson, S. Gallivan and J. Burridge, "Estimating a Markov Transition Matrix from Oberservational Data," *J. Oper. Res. Soc.*, vol. 46, no. 3, March 1995.

[19] S. Shetty, M. Song, C. Xin and E. K. Park, "A Learning-based Multiuser Opportunistic Spectrum Access Approach in Unslotted Primary Networks," *in Proc. of IEEE INFOCOM'09*, Rio de Janeiro, Brazil, April 2009.

[20] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction", the MIT press, 1998.

[21] C. Tekin and M. Liu, "Online Learning in Opportunistic Spectrum Access: A Restless Bandit Approach," *in Proc. of IEEE INFOCOM'11*, Shanghai, China, April 2011.

[22] C. Tekin and M. Liu, "Approximately Optimal Adaptive Learning in Opportunistic Spectrum Access," *in Proc. of IEEE INFOCOM'12*, Orlando, FL, March 2012.

[23] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized Cognitive MAC for Opportunistic Spectrum Access in Ad Hoc Networks: A POMDP Framework," *IEEE J. Sel. Areas. Commun.*, vol. no. 3, April 2007.

[24] Q. Zhao, S. Geirhofer, L. Tong and B. M. Sadler, "Opportunistic Spectrum Access via Periodic Channel Sensing," *IEEE Trans. Signal Process.*, vol. 56, no. 2, Feb. 2008.