Channel Sensing Order in Multi-user Cognitive Radio Networks

Jie Zhao and Xin Wang
Department of Electrical and Computer Engineering
State University of New York at Stony Brook
Stony Brook, New York 11794
{jiezhao, xwang}@ece.sunysb.edu

***Abstract*—This paper investigates the sensing order problem for multi-user and multi-channel cognitive radio networks. While most of the literature studies focus on the sensing order for a single user, we consider the scenario in which multiple secondary users sequentially sense and access the channel according to their individual sensing orders. In multi-user case, channel access collisions among secondary users will lead to performance degradation. We propose a novel metric that comprehensively consider the channel availability, transmission rate and collision probability, and exploit an efficient dynamic programming algorithm to establish the sensing order based on the metric to improve the sensing efficiency and transmission throughput. Simulation results demonstrate that our algorithm can not only effectively reduce the collisions among secondary users, but also can achieve higher network throughput than other schemes studied. Furthermore, we also discuss how the network environment impacts the performances of different sensing orders.**

***Index Terms*—opportunistic spectrum access; cognitive radio; multi-user sensing order; dynamic programming.**

I. INTRODUCTION

There are growing research interests on cognitive radio networks in recent years [1]–[4]. Existing works generally assume a secondary user (SU) has a number of potential wireless channels, and a SU can randomly select a channel to sense or sense the channel sequentially according to a predefined sequence [5]. An SU can transmit data if the channel is detected to be free from the PU occupancy. This random sensing or simple sequential sensing, however, could be very inefficient. On one hand, if a secondary user senses a channel often occupied by the primary users, it is most likely that the secondary user will find this channel to be busy (PUs are present). The user has to give up the sensed channel and switch to another channel to sense. This will not only require additional sensing time but also incur non-negligible switching delay between channels, which would lead to a high sensing overhead and lower transmission efficiency. These in turn would result in higher transmission delay and lower throughput. On the other hand, if a secondary user senses a channel that is rarely used by the primary users, it will probably find this channel to be free from primary activities and start to use this channel.

The channel sensing sequence taken by a secondary user will significantly influence the effectiveness of channel discovery and utilization. Therefore, it is important for the SUs to determine the channel sensing order according to which they could sequentially sense the channels. [6] denoted this as a sensing order problem and have proposed some strategies to design the channel sensing order for a single secondary user. Besides channel availability, it is also important to look for a channel that has good condition to support higher data rate. Sensing ordering for single user primarily based on the data rate has also been studied [7].

However, in a general multi-user network, the channel utilization will not only be influenced by the activities of primary users and channel achievable rates, but also the mutual impacts among secondary users. We refer to the conflicting interests of multiple users on the same channel as “collision”. The collision includes not only the situation that at least two secondary user pairs sense and transmit over the channel at the same time (we call it “channel contention”), but also the circumstance that a secondary user senses a channel that is being used by another secondary user (we call it “useless sensing”). If a collision occurs, a secondary user would waste an opportunity of discovering the available channel. The consequent additional sensing overhead would result in the throughput degradation of both the individual users and the overall network. Although important, there are very limited studies on the impact of collision on the sensing order. The work in [8] constrained the sensing order problem to two-user multichannel cognitive medium access control, with the emphasis on the contention-resolution strategies. It remains a challenge for each user to find an efficient sensing order distributively to reduce the collision in a network with multiple users.

From above discussions, we can see that the sensing order performance is impacted by three major factors: channel availability, channel rate, and the possibility of channel access collision. Since each factor is important, it is needed to concurrently consider these three factors in determining the sensing order. In this work, we propose a novel sensing metric that can coherently integrate the three factors to guide the sensing order finding, and a dynamic programming algorithm that allows each node to efficiently determine its sensing order in coordination with neighboring nodes. Simulations show that our scheme can not only reduce the collisions among different users but also can achieve better secondary network throughput.

The rest of this paper is organized as follows. The next section gives an overview of related work. We then discuss the various issues to consider for determining the sensing order in a network with multiple secondary users in Section III, and provide the system model in Section IV. In Section V, we introduce the metric to measure the reward of sensing a channel jointly considering all three major factors that impact sensing ordering discussed above, and exploit a dynamic programming algorithm for search of the sensing order distributively. Section VI presents the simulation results and performance analysis, and section VII concludes this paper.

II. RELATED WORK

In the literature, the optimal sensing order problem is discussed mostly for a single user.

In [6], H. Jiang et al. investigate the sensing order problem for multi-channel cognitive medium access control with opportunistic spectrum access. One contribution of this work is to show that although the Intuitive Sensing Order based on the descending order of the channel primary-free probabilities is optimal when the transmission rate is fixed without adaptation, it does not lead to the optimality in general with adaptive modulation and rate. This paper also proposes a dynamic programming methodology to search for single-user sensing order by considering both channel availabilities and maximum achievable data rates. We refer to this scheme as DP.

Similarly, H. Kim et al. [10] address the problem of rapid spectrum opportunity discovery for seamless service provisioning for secondary users (SUs) in cognitive networks. It proposes an efficient sensing sequence (sensing order) that incurs a small opportunity-discovery delay. To support the proposed sensing-sequence, they present a channel-management strategy that optimally selects and updates the list of backup channels and also provide methods for flexible estimation of ON/OFF channel-usage patterns and prediction of channel availability.

In [7], a simple channel order for SUs without *a priori* knowledge of primary user activities is proposed. By sensing the channels according to the descending order of their achievable rates and selecting the first available (sensed free) channel, it shows that the proposed channel exploitation approach is efficient yet effective in increasing the throughput and resource utilization. We refer to this scheme as RATE.

The optimal stopping problem is usually associated with sequential sensing, therefore, the stopping strategy is also a significant element in sensing order problem. T. Shu et al. in [9] derive the throughput-optimal decision strategy for the sequential channel sensing/probing process. The proposed use-or-skip decision strategy maximizes the CR's average throughput using the optimal stopping theory. Rather than looking for an optimal stopping rule, in our work, we aim to design a more efficient sensing order which various algorithms on stopping rules can leverage. With a more efficient sensing order, however, it obviates the need for a complex stopping rule.

While most of the literature studies focus on determining the sensing order for a single SU, some problems that remain to be resolved are: 1) Can the sensing order established for a single user be directly applied to multi-user case? 2) How to reduce collisions among users to reduce performance degradation? 3) How to design the sensing order for each secondary user distributively?

Closely related to our work is [8], which has discussed the sensing order problem in the case of two-user multi-channel cognitive medium access control, with the emphasis on the contention-resolution strategies. The major differences between our work and [8] can be explained as follows. [8] is limited to two-user case and requires a coordinator in the network to determine the sensing orders for each of the two users. However, our algorithm is designed for an arbitrary number of users in a distributed network with no coordinator, and every user can obtain its unique sensing order by coordinating with its neighbors. In addition, we propose a novel metric to comprehensively incorporate all major factors that impact sensing efficiency, and a dynamic programming algorithm to efficiently search for the sensing order.

III. PROBLEM AND MOTIVATION

As discussed in Section I, three major factors need to be considered when determining the sensing order in a multi-user network, including the channel availability, the channel achievable data rate, and the possibility of channel access collision among secondary users. First, in cognitive radio networks, it is important to detect an idle channel that is not occupied by primary users to prevent interfering with PUs. Without considering the channel availability for secondary use, it may increase the sensing time and result in delayed transmission and lower data throughput. Second, in a wireless environment, different channels could have different interference and fading conditions. If a channel detected free has a bad channel condition, the transmission rate of the channel would be very low. Third, in a network with multiple users, it is critical to consider the inefficiency as a result of "collision" (channel contention and useless sensing mentioned in Section I) among secondary users on the same channel. The main difference between our work and the other literature work is that our sensing order design concurrently considers all three factors, and pays special attention to reducing sensing collisions among multiple secondary users. Since the literature work generally study sensing order problem for a single user without considering the sensing collision issue, we will discuss more on the negative impact of this problem in this section.

As mentioned in Section I, secondary user collision includes two cases, one is channel contention due to simultaneous detections or transmissions from more than one user on the same channel, and the other is useless sensing when a user senses the channel occupied by another secondary user already. Although a transmission pair may reach the agreement of a sensing order, they may not tune to a channel on the list at exactly the same time. In order to facilitate the transmissions between the sender and receiver, a short handshake message

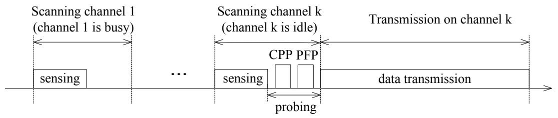

Fig. 1. Sequential channel sensing and probing before transmission when the user stops at the kth scanning. If the channel is sensed busy, it will not be probed. If the channel is sensed free, channel probing will be conducted through CPP (Channel Probing Packet) and PFP (Probing Feedback Packet).

may be sent before the data transmissions. The preamble of the message can be applied to estimate the channel conditions and the pair of messages can serve as probing purpose [9]. As is shown in Figure 1, the transmitter first sends a Channel Probing Packet (CPP) to the receiver through the channel it senses free after a small random delay. If the receiver successfully receives this message and also senses the channel free, it can send back a Probing Feedback Packet (PFP) carrying the information of the maximum achievable transmission rate of this channel. In case that multiple users send the probing messages at the same time, there will be a collision. In addition, the receiver may sense the channel busy (due to the PU's or other SU's activities). In either case, the sender cannot receive the feedback from the receiver. If collided users give up the channel and continue to sense and probe other channels, the current channel opportunity, if any, will be wasted and additional sensing delay will be introduced to find a new channel. If a sender retransmits the CPP message, additional delay would still be introduced, which explains the first case of collision. Both collision cases will introduce extra overhead.

To further explain when a collision can happen, let's take a look at a simple 2-by-3 case. There are 2 secondary users (SU_1, SU_2) and 3 channels (1,2,3), and we can see how the sensing orders affect the channel utilization significantly. For SU_1 and SU_2, if we set their sensing orders to be the same, there is a high probability of probing collision (channel contention). If the sensing orders are different in some positions, there might still be probing collisions. For example, for SU_1 if the sensing order is $(1, 3, 2)$ and the sensing order for SU_2 is $(2, 3, 1)$, if both channel 1 and channel 2 happen to be occupied by the primary users, SU_1 will sense the channel 1 to be busy, SU_2 will sense the channel 2 to be busy (if the sensing result is always right), so both users will continue to sense 3. If they both find channel 3 to be free and probe that channel, their probing attempts may collide with each other.

We can see that the collisions are associated with the sensing orders. It remains a challenge for each user to find a sensing order to reduce the sensing collisions with other secondary users while ensuring timely detection of an idle channel at a higher transmission rate.

IV. SYSTEM MODEL

We consider a cognitive radio network with N secondary users with indices from 1 to N, M potential data channels with indices from 1 to M, and one common control channel. Each secondary user is able to determine its own sensing order, according to which the user can sequentially sense one channel at a time until it finds an available channel to transmit data. Assume the channel i ($i \in \mathcal{M} = \{1, 2, ..., M\}$) is free from primary activities with a long-term probability $\theta_i \in [0, 1]$, which is the availability of that channel. There is no correlation between the primary activities in different channels, i.e., for each channel, the primary-free probability is independent of those in other channels.

We also consider a time-slotted structure shown in Fig. 2, which consists of two kinds of phases, a negotiation phase and a sensing/probing/transmission phase. Within a sensing/probing/transmission phase (we call it a slot in this paper) as shown in the figure, a user will sequentially sense and probe each channel according to a sensing order until it finds an idle channel not occupied by the primary users or other secondary users and transmit data in the remaining time of the sensing and transmission duration. If a user has some specific requirements on the transmission channel, it can determine if it will take the channel sensed or continue to detect the next channel based on the channel condition. For example, if the user has a certain transmission rate requirement and the rate estimated is lower than its expectation, it will give up the current channel and continue to sense the next channel on the list of the sensing order.

If a user decides to access a channel, we say that the user stops at that channel. We consider that no recall or guess is allowed, i.e., users can't go back to sense the channels that have been sensed, and they can't guess the status of a channel (busy or idle) without actually sensing it. Fig. 2 shows that a user stops after its kth sensing and starts transmission until the end of the slot. The average time taken to scan (sense and probe) a channel is denoted by τ, and the time length for a sensing/probing/transmission phase is denoted by T. Therefore, each slot consists of a sensing phase (no longer than $M\tau$) and a transmission phase (no shorter than $T - M\tau$). If a user stops at the kth channel in its sensing order, the lengths of the sensing process and transmission process are $k\tau$ and $T - k\tau$, respectively. The *effectiveness* of a slot is defined as the ratio of the transmission phase length to the slot length. Therefore, if a user stops at the kth channel in its sensing order, the effectiveness is $c_k = 1 - k\tau/T$.

In the negotiation phase, a transmitter node will determine its sensing order in coordination with its target receiver and other neighboring nodes. Specifically, every transmitter first obtains a Preliminary Sensing Order (PSO) by sorting all the channels according to the descending order of their achievable rates estimated from past measurements. The purpose of using the rate to guide the preliminary ordering is to allow each user to maximize its chance of getting the higher rate channel. Neighboring nodes will carry out the negotiation over the common channel, along with their handshake messages which are needed to coordinate their synchronization and transmissions. A transmitter and receiver pair will negotiate and determine the PSO that both can agree to. Based on its knowledge of the

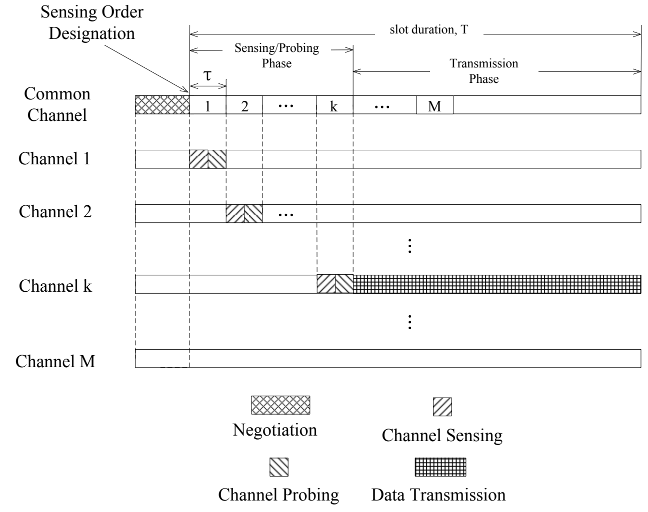

Fig. 2. Slotted sensing/probing and transmission structure (when a secondary user stops after its kth scanning).

PSO of neighbors overheard from their negotiations and also taking into account the channel availability and the need of reducing secondary sensing collisions among neighbors, at the beginning of the sensing and transmission phase, a transmitter will take the Sensing Order Designation step to determine the Actual Sensing Order (ASO) that they will actually execute. The algorithm for reducing the sensing collision in Sensing Order Designation will be described in Section V.

In our current design, we obtain the PSO by sorting the channels according to the descending order of their estimated achievable rates. There are also some other options for designing the Preliminary Sensing Order, e.g. DP and RATE proposed in [6] and [7], respectively. In Section VI, we will investigate how different PSOs can affect the ASO performances.

In order to reduce the negotiation overhead, it is possible to increase the length of the sensing and transmission slot, or associate multiple slots with one negotiation period, i.e. the users negotiate and updates their sensing orders every multiple slots. In the first case, once determining a channel to scan, a user will sense before each transmission to avoid interference and collision with both primary users and secondary users. Periodic and relatively longer period in-band sensing [10] could also be arranged to better prevent interfering with primary users. In the second case, users are able to periodically perform sensing order updates to find a better channel, in which case sensing order may be adjusted based on the sensing and transmission results from previous slots. These options are not our major focus.

In a slot (sensing and transmission phase), a secondary transmitter and receiver pair executes the following steps:

A. Sensing Order Designation

According to the PSOs of other neighboring users overheard in Negotiation Phase, all users are able to design a new Actual Sensing Order (ASO), which they will follow to actually sense the channel. The algorithm to determine this ASO will be discussed in the next section.

B. Sensing, Probing and Transmission

According to the Actual Sensing Order, each transmitter first senses a channel, if this channel is busy, it continues to sense the next channel in the order. If a channel is detected idle, it sends a Channel Probing Packet (CPP) to the receiver over the idle channel. After the receiver gets the probing message, it can estimate the maximum achievable data rate over this channel for the user pair. If the receiver also senses the channel to be free, it will confirm the use of this channel by sending back a Probing Feedback Packet (PFP) to the transmitter over the same channel attaching the channel rate estimated [9]. The transmitter then also knows the achievable rate over this channel and will start to communicate with the receiver over this idle channel till the end of the transmission phase.

A receiver will not send back a Probing Feedback Packet in some cases: 1) The receiver tunes to the channel late and misses the CPP message, 2) The receiver senses the channel to be busy, and 3) Probing collision happens among multiple transmitters. If the transmitter does not receive the PFP message for a threshold time, it will retransmit CPP message several times, e.g. three, each with some random back-off time, before tuning to the next channel to sense.

As mentioned in previous sections, channel condition (availabilities, maximum achievable rates) information is needed to design an efficient sensing order. Here are some options for a node that newly enters a neighborhood without the knowledge of the primary-free availability and the rate of a channel. It can assign the channel with an initial primary-free availability, which can be set as the average value of the primary-free availability of neighbors. Each channel can be assigned the same random rate for equal chance of selection. The rate of the channel will be first set to its rate value obtained through a successful probing for the first time, and updated each new rate is obtained. The primary-free availability and rate of the channel will be updated to more precise values as the node senses and probes the channel. The estimation of the channel availability and finding of the maximum achievable channel rate have been studied the literature work, and this problem will not be our main focus in this paper. For the channel availability estimation, H. Kim et al. in [10] propose a Bayesian estimation technique which can perform reasonably well if the number/frequency of samples is limited, which fit well for our case because in our methodology some channels might not be often visited. In [11], S. Yin et al. exploit spectrum correlation to develop a 2D frequent pattern mining algorithm that can predict channel availability based on past observations with considerable accuracy. For the estimation of the maximum channel achievable rate, A. Katidiotis et al. in [12] develop a learning scheme that relies on artificial neural networks to estimate the achievable transmission data rate.

V. MULTI-USER SENSING ORDER SELECTION

In this section, we will introduce the algorithm for each secondary user to select the sensing order for reducing the sensing delay and increasing the transmission throughput. Users in the neighborhood will cooperate in determining the sensing order to reduce their sensing collisions, thus improving the performances of the secondary user network.

We have three major factors to consider in the sensing order selection, channel availability, channel maximum achievable rate and channel collision probability. We consider both the channel availability (i.e., free from primary use) and the channel collision-free probability (i.e., free from secondary use) to calculate the *Channel Free Probability*, which is impacted by both the primary activities and secondary activities. The Channel Free Probability is thus determined by the channel availability for the secondary use and the probability of SU collision on a channel. The Channel Free Probability is calculated based on the Preliminary Sensing Orders discussed in the previous sections.

We first introduce the methods of calculating the Channel Free Probability in two-user case and multi-user case respectively, and then adopt a dynamic programming scheme to search for the sensing order based on all three factors.

A. Channel Free Probability in Two-user Case

We first start from 2-user case, i.e., $N = 2$, and we denote the two secondary users as SU_1 and SU_2 respectively. Each user has a Tentative Sensing Order, i.e., $\mathcal{A} = (a_1, a_2, .., a_M)$ for SU_1 and $\mathcal{B} = (b_1, b_2, ..., b_M)$ for SU_2, which are permutations of channel indices $(1, 2, .., M)$. Note that a_k and b_k denote channel indices.

The Channel Free Probability incorporates the primary-free probability and collision-free probability. During the negotiation phase, every user is able to know other neighboring users' Preliminary Sensing Order, which will be used to calculate the Channel Free Probability.

The $sensing\ position$ of a channel associated with a user is defined as the position of the channel in the user's sensing order. In the two-user case, channel a_k's (or b_k's) sensing position with SU_1 (or SU_2) is k. If channel i is in an order $\mathcal{O}$, $p(i, \mathcal{O})$ is used to denote the position of i in $\mathcal{O}$. Then for a channel $i \in \mathcal{A}$ of SU_1, we have the following Channel Free Probability:

$$\theta_{i,1}^{free} = \theta_{i,1}(1 - \prod_{j=1}^{p(i,\mathcal{B})-1} (1 - \theta_{j,2})) \tag{1}$$

where $\theta_{i,n}$ denotes the primary-free probability of channel i (i is the channel index) for SU_n, and $(1 - \prod_{j=1}^{p(i,\mathcal{B})-1}(1 - \theta_{j,2}))$ is the probability that SU_2 does not proceed to sense channel i in its own sensing order $\mathcal{B}$. SU_2 senses the channel i only if all channels before i in its sensing order $\mathcal{B}$ are busy.

Similarly, for a channel $i \in \mathcal{B}$ of the user 2, we have the following to get the Channel Free Probability for SU_2:

$$\theta_{i,2}^{free} = \theta_{i,2}(1 - \prod_{j=1}^{p(i,\mathcal{A})-1} (1 - \theta_{j,1})). \tag{2}$$

The primary-free probability of channels will be attached when a user transmits the Preliminary Sensing Order.

B. Channel Free Probability for More Than 2 User Case

When there are more than 2 secondary users in the network, the determination of channel free probability is similar. If there are N users, then a user needs to consider the possibility of collision with the other $N - 1$ users.

Let the sensing order for user $SU_1, SU_2, ..., SU_N$ be $\mathcal{O}_1, \mathcal{O}_2, ..., \mathcal{O}_N$, respectively. For any user SU_k ($k = 1, 2, ..., N$) and the user index set $\mathcal{N} = (1, 2, ..., N)$, the calculation of the channel free probability for a channel $i \in \mathcal{O}_n$ at the nth user SU_n is:

$$\theta_{i,n}^{free} = \theta_{i,n} \prod_{k=\mathcal{N}\backslash n} (1 - \prod_{j=1}^{p(i,\mathcal{O}_k)-1} (1 - \theta_{j,k})) \tag{3}$$

where $\theta_{i,n}$ denotes the primary-free probability of the channel i for SU_n, $(1 - \prod_{j=1}^{p(i,\mathcal{O}_k)-1}(1 - \theta_{j,k}))$ is the probability that the secondary user SU_k does not proceed to sense channel i, and $\prod_{k=\mathcal{N}\backslash n}(1 - \prod_{j=1}^{p(i,\mathcal{O}_k)-1}(1 - \theta_{j,k}))$ is the multiplication of all the individual probabilities of every user except SU_n not proceeding to sensing channel i, which is probability that all the other $N - 1$ users (except SU_n) will not proceed to sense the channel i.

Each user calculates the Channel Free Probabilities for every channel independently. We now present our algorithm which makes use of this Channel Free Probability to design the sensing orders.

C. Dynamic Programming Search

We propose a dynamic programming approach for each secondary user to find an Actual Sensing Order, based on both the Channel Free Probability and the channel maximum achievable rate. Throughout this paper, if not otherwise specified, when we say channel order/sensing order, it refers to the Actual Sensing Order. As discussed in the previous section, the Channel Free Probability also depends on both the availability of the channel without primary occupancy and the probability of collisions among the secondary users. Therefore, our channel order search considers all the three major factors we introduced earlier.

To guide the dynamic programming process, there is a need to find a reward function. We define the reward for a channel i in an order $\mathcal{O}$ as its effective throughput as follows:

$$U = \theta_i^{free} c_{p(i,\mathcal{O})} R_i, \tag{4}$$

which depends on the probability of the channel i being free from both primary user and other secondary users. Recall that $p(i, \mathcal{O})$ is used to denote the position of i in $\mathcal{O}$. If a user stops at the $p(i,\mathcal{O})$th channel in its sensing order, the channel effectiveness factor $c_{p(i,\mathcal{O})}$ equals $1-p(i,\mathcal{O})\tau/T$.

In order to find the sensing order that leads to the maximum reward, the dynamic programming search for a sensing order starts from the last position of the order, which is the opposite of the sensing sequence. For example, if a preliminary sensing order contains five channels $\mathcal{O}=\{i_1,i_2,i_3,i_4,i_5\}$, we will determine the channel at the position i_5 first. In the process of searching for the maximum reward of a sensing order, the cumulative reward will be calculated as shown in each step below:

Stage 1 – Calculate the maximal expected reward value associated with the last channel in the sensing order. In this stage, if all M channels need to be sensed, a state is indicated by the set of the previous $M-1$ channels in the sensing order $(i_1,i_2,...,i_{M-1})$. So there exist $\binom{M}{M-1}$ states. For a certain state $(i_1,i_2,...,i_{M-1})$, the maximal reward is

$$J^1_{max}(i_1,i_2,...,i_{M-1}) = \theta^{free}_{l\in\mathcal{M}\setminus(i_1,i_2,...,i_{M-1})} c_M R_{i_M} \tag{5}$$

where the superscript "1" represents the stage number (1 in this case) and $\mathcal{M}$ means the set of all M potential channels, R_i indicates the maximal achievable data rate over channel i.

Stage k – Calculate the maximal cumulative reward value backwards from the last channel of the sensing order until the channel at the kth position from the last, based on results in state $k-1$. At the kth stage, $M-k$ channels, denoted by $(i_1,i_2,...,i_{M-k})$, have not been assigned in the sensing order. Hence there are $\binom{M}{M-k}$ states. The lth $(1\leq l\leq k)$ transition leads to the state $(i_1,i_2,...,i_{M-k},j_l)$, a state at the stage $k-1$, which means the channel $j_l\in\mathcal{M}\setminus(i_1,i_2,...,i_{M-k})$ and the channels $\{i_1,i_2,...,i_{M-k}\}$ have not been assigned in the sensing order. The cumulative reward associated with the lth transition is denoted as

$$\begin{aligned} &U(j_l, J^{k-1}_{max}(i_1,i_2,...,i_{M-k},j_l)) \\ &= \theta^{free}_{j_l}(c_{M-k+1}R_{j_l}) + (1-\theta^{free}_{j_l}) J^{k-1}_{max}(i_1,i_2,...,i_{M-k},j_l) \end{aligned} \tag{6}$$

So the maximal reward of state $(i_1,i_2,...,i_{M-k})$ is

$$\begin{aligned} &J^k_{max}(i_1,i_2,...,i_{M-k})) \\ &\quad = \max_{1\leq l\leq k} U(j_l, J^{k-1}_{max}(i_1,i_2,...,i_{M-k},j_l)) \\ &= \max_{j_l\in\mathcal{M}\setminus(i_1,i_2,...,i_{M-k})} U(j_l, J^{k-1}_{max}(i_1,i_2,...,i_{M-k},j_l)) \end{aligned} \tag{7}$$

In each stage k, the channel index j_l is

$$j_l = \underset{j_l\in\mathcal{M}\setminus(i_1,i_2,...,i_{M-k})}{\arg\max} U(j_l, J^{k-1}_{max}(i_1,i_2,...,i_{M-k},j_l)) \tag{8}$$

then j_l in the stage k will be recorded as the $(M-k+1)$th channel index in the sensing order.

At each state, the channel index associated with the transition that leads to the maximal reward will be recorded. After the maximal reward value is obtained at the stage M (i.e., the stage when no channel has been sensed yet), an sensing order can be traced back according to the recorded optimal channel selection at each state. Instead of putting all possible channels in the sensing order list, if only L out of M channels will be sensed, the algorithm above can be similarly applied.

When there are four potential channels in the network, i.e., $M=4$, the proposed dynamic program search can be shown in Figure 3. Here all the four channels are in the sensing order. In Stage 1, the last (4th) channel in the sensing order is determined, and in Stage 2, the 3rd channel in the sensing order is determined, and Stage 3 the 2nd channel. Consequently, the first channel in the sensing order is determined in the last step (Stage 4). Take the red-line trace as an example. It shows that in Stage 1, the dynamic programming algorithm indicates that the expected reward value is the largest when channel 2 (i.e., the one not included in the oval) is set as the last channel in the sensing order, therefore the last channel is determined as channel 2. Then in Stage 2, the algorithm finds that the expected reward value is the largest when channel 3 is set as the 3rd channel in the sensing order. At this time, the last two channels in the sensing order have already been determined. Similarly, in Stage 3, channel 1 is determined as the 2nd channel in the sensing order and in Stage 4 channel 4 is set as the first. Thus, by tracing backwards from Stage 4 to Stage 1 we can get the sensing order as {4,1,3,2}.

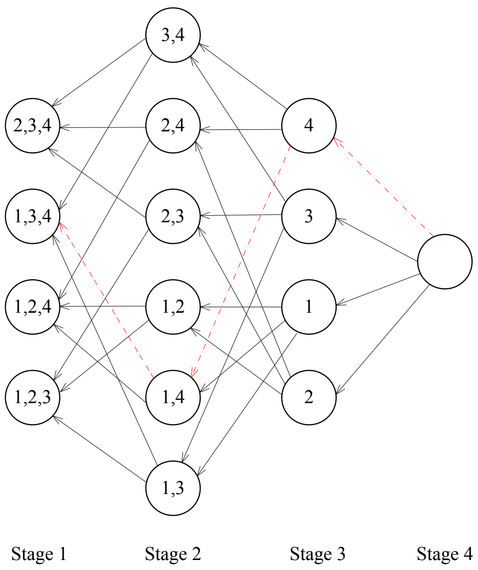

Fig. 3. The dynamic programming process with M=4. The red-dashed trace indicates the sensing order as {4,1,3,2}.

Instead of purely based on dynamic programming, some adjustments of the sensing order may improve the performances. For example, the first channel in the sensing order may be set to the one used in the previous slot given that it is very likely that it will also be available in this time slot. If the sensing order is always based on the dynamic programming, some channels at the end of the list may not get chance to be sensed and probed, thus leading to lack of information. To mitigate this problem, every user has a small probability of choosing a random order so that the information of some new channels can be obtained. If the probability is high, it would enable more active sensing of new channels for possible system-wide performance improvement. On the other hand, with a small probability of random channel sensing, the user could take our proposed sensing order settings to maximize its performance based on existing knowledge.

Here we demonstrate an example algorithm for each user to find the sensing order, which can be implemented according to actual applications. The ϵ-greedy algorithm is shown in Algorithm 1.

Algorithm 1 ϵ-greedy Algorithm in a slot

```
for all slots do
    Choose a small probability $\epsilon$ between 0 and 1
    Generate a random number $r$ between 0 and 1
    if $r < \epsilon$ then
        select a random sensing order
    else
        select the sensing order calculated by the proposed Dynamic Programming Search.
    end if
    Move the channel used successfully (if any) in the previous slot to the first place of the sensing order found so far.
    Execute the new sensing order and record the channel that has been successfully used.
end for
```

VI. SIMULATIONS AND RESULTS

In this section, we will demonstrate the performances of our proposed sensing order. We will compare our schemes with the random sensing order and the proposed sensing orders in [6] and [7]. To make this comparison, we directly apply these sensing order (proposed for single user) to multi-user case, i.e., every secondary user executes the single user sensing order regardless of the decisions of other users. In contrast, our proposed sensing order designation has considered the possible activities of the other secondary users to avoid collisions and improve the sensing efficiency. As we will see, our proposed algorithm for sensing ordering can both improve the overall SU network throughput and reduce collisions among SUs.

A. Simulation Settings

In our simulations, we consider a SU network with M potential channels and N secondary users. If not otherwise mentioned, we adopt the default values as $M = 9$, $N = 5$.

The PU activity for each channel is modeled by an ON/OFF process with ON or OFF period exponentially distributed. An ON period means the primary user is present in the channel, thus it cannot be used by the secondary user. An ON period means the primary user is present in the channel, thus it cannot be used by the secondary user. Therefore, the channel availability is determined by the total time duration of OFF periods divided by the time duration of all (ON and OFF) periods. If not otherwise specified, the default value of the average channel availability among all the channels is set to 0.7.

In our simulations, secondary users are randomly distributed in an area. Due to different channel gains, the achievable rates of the available channels for one secondary user pair are plausibly different from those of the other user pairs. We use the channel rate variability to indicate this difference. The default channel rate variability is set to 5, which means that the maximum achievable rate perceived by different users are 5 units (e.g. kb/s) from each other. The larger the variability, the channel rates of a channel for different users are more distinct. In reality, if all the users are very far away from each other, they may have very different achievable rates for the same channel. The SUs sequentially access channels according to their sensing orders and start transmission once they find an idle channel. If a collision (channel contention and useless sensing, discussed in Section III) occurs, the colliding users may need to retry for the channel through some collision-resolution strategies or move to sense another channel depending on if the collision is due to channel contention or useless sensing.

To be general and reduce the effects of simulation randomness, we average the performance over 1,000 runs, where each run contains 100,0000 sensing/probing/transmission phases (slots). To reduce the overhead of negotiation phase in Fig. 2, we consider that the negotiation phase occurs every 1000 slots so that every user can exchange some information, such as the channel availability detected by each users. The sensing order, however, will be updated more frequently as the channel rate sensed by a user changes. Other system parameters are sensing/probing time $\tau = 50\mu s$ and slot duration $T = 50ms$. We then average over the simulation runs to demonstrate the performances.

We consider two major performance metrics. One is the normalized throughput, which equals the actual throughput of all the secondary users divided by the maximum achievable throughput of all the secondary users; the other is called collision probability, which equals the number of collisions of all the users divided by the number of all scanning (sensing/probing) attempts of all the users. Obviously, a good sensing order should achieve both high normalized throughput and low collision probability. In the simulations we are going to investigate the impacts of PSOs, the number of channels, the number of secondary users, the average channel availability and the channel rate variability (how much different the channel rates are across all the channels) on the performances of the sensing orders.

We compare the performance of our sensing order with that of three reference schemes, e.g. RATE (which sorts the channels according to the descending order of the maximum achievable rates) proposed in [7], DP (which considers both channel availability and transmission rate and uses Dynamic Programming search for single-user sensing order) in [6] and RANDOM (with which each user randomly generates its sensing order). We will not investigate the Intuitive Sensing

Order mentioned in [6], which sorts the channels according to the descending order of channel availabilities. In multi-user case, the neighboring nodes will perceive very similar primary-free probabilities (i.e. availabilities) for every channel, if every user adopts the Intuitive Sensing Order, the collision probability among SUs will be very high because their sensing orders are very similar. In this case the Intuitive Sensing Order is not suitable for multi-user case and might not even be able to compete with RANDOM.

B. Effect of Preliminary Sensing Order

As discussed in Section V, the choice of Preliminary Sensing Order (PSO) will impact the performances of our proposed sensing order determination algorithm. Before evaluating the performance of our complete algorithm and comparing our algorithm with peer ones, we'll first evaluate the impact of three PSOs on our algorithm. The first one, PSO-RATE, sorts all the channels according to the descending order of their maximum achievable rates as proposed in [7]; the second one, PSO-DP, obtains the PSO as the sensing order settings proposed in [6]; the third one, PSO-RANDOM, gets the channel order randomly. Note that PSOs are not the final sensing order.

In Fig. 4 and 5, the normalized throughput and sensing collision probability for $N = 4$ users are depicted for different PSOs. As expected, when the number of channels increases, the normalized throughput of all PSOs increases while the collision probability among SUs reduces. With more channels, the users have more choices in sensing and channel selection and are less likely to collide with each other, which help to increase the throughput.

Among all the three PSOs, PSO-RATE can achieve the lowest collision probability and the highest normalized throughput. This sensing order selection not only benefits each user with a higher rate channel, but also helps to reduce the collision among user channel sensing. This is because different users will usually perceive different channel rates, and if each user tries to sense its best channel first, the chance of multiple users sense the same channel will reduce. PSO-RANDOM performs the worst in both performance as it neither considers how to reduce the collisions nor how to improve the throughput. PSO-DP benefits from the consideration of rate, but its consideration of availability will reduce the randomness in determining the sensing order of neighboring users, thus leading to higher collision and reduced throughput as compared to those of PSO-RATE. According to previous discussions, in the rest of our simulations, we choose PSO-RATE as the PSO of our proposed sensing order.

C. Effect of the number of channels

For $N = 5$ users, the normalized throughput and collision probability for different sensing orders are depicted in Fig. 6 and Fig. 7, respectively. We can see that our proposed sensing order always outperforms other sensing orders, with higher normalized throughput and lower collision probability.

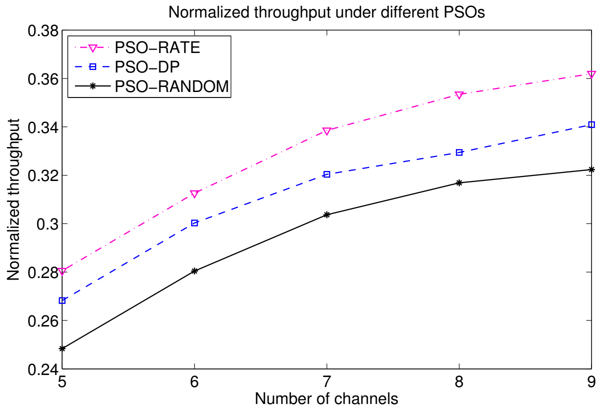

Fig. 4. Normalized throughput with different PSOs when N=4 (4-user case).

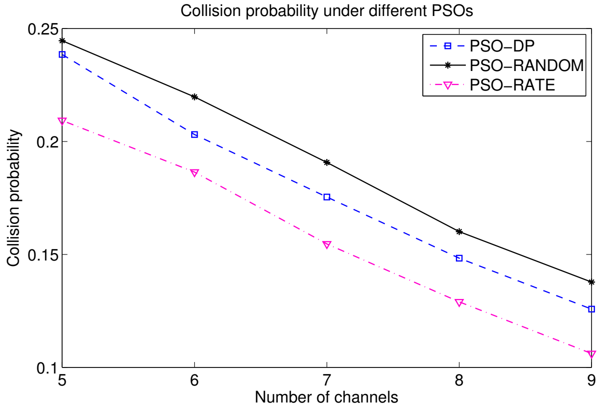

Fig. 5. Collision probability with different PSOs when N=4 (4-user case).

For all the sensing orders, as the number of channels increases, the normalized throughput increases and the collision probability decreases. This can be explained as follows: if there are more channels in the network, the users are less likely to collide and will have more efficient channel utilization.

Also, the advantage of our algorithm is more obvious when the number of channels is small. When there are $M = 5$ channels, in terms of the normalized throughput, our proposed sensing order can outperform RATE, DP and RANDOM by approximately 70%, 70% and 150%, respectively, and when $M = 9$, these numbers become 7%, 50% and 55%. When the number of channels is small, the collision probability is higher. This further demonstrates that it is very important to take into account the probability of sensing collision when establishing the sensing order.

D. Effect of the number of secondary user pairs

We will further investigate how the number of user pairs affects the performances of sensing orders. In Fig. 8 and Fig. 9, it can be seen that the proposed sensing order always performs better than other orders. As expected, when the number of user pairs (N) increases, the collision probability increases. Consequently, the normalized throughput decreases.

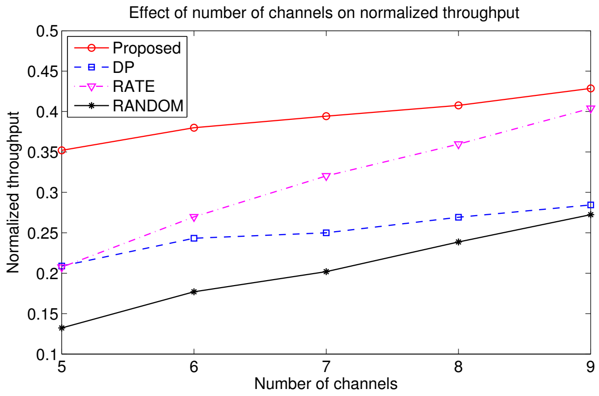

Fig. 6. Effect of number of channels on normalized throughput.

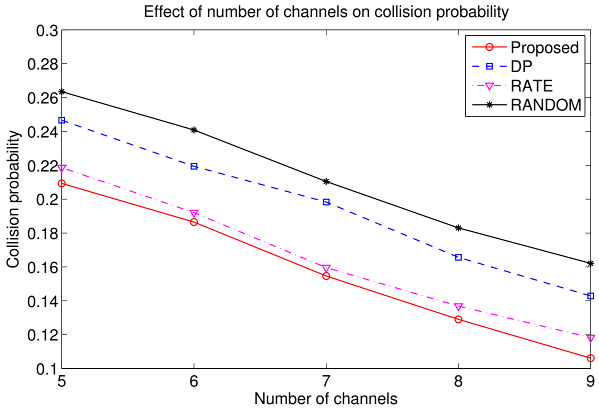

Fig. 7. Effect of number of channels on collision probability.

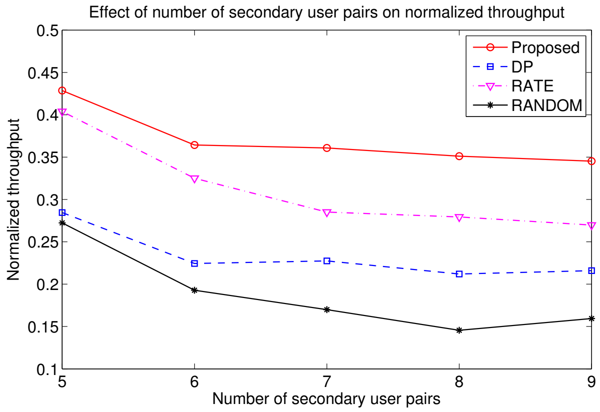

Fig. 8. Effect of number of users on normalized throughput.

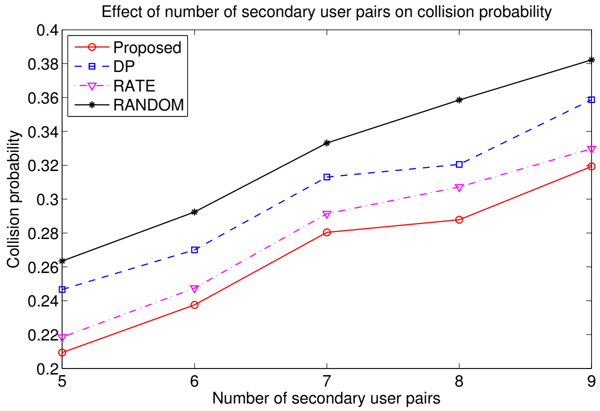

Fig. 9. Effect of number of users on collision probability.

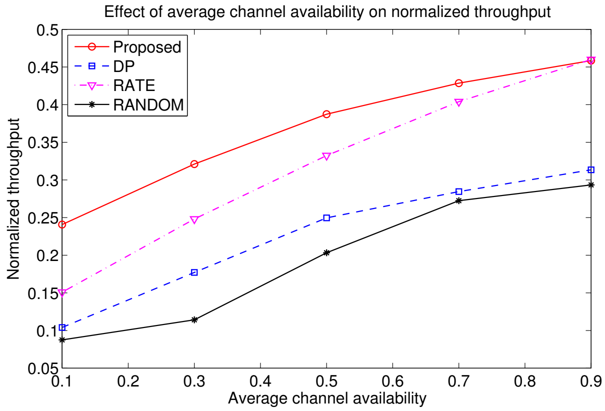

Fig. 10. Effect of average channel availability on normalized throughput .

When there are $N = 5$ user pairs, in terms of normalized throughput, our proposed sensing order can outperform RATE, DP and RANDOM by approximately 6%, 50% and 55%, respectively. These numbers increase to 27%, 67% and 118% as the number of users increases further to $N = 9$. Again, the performance improvement increases as the number of users increases, because our proposed algorithm can effectively reduce the sensing collision to increase the throughput. These results are obtained when $M = 9$ channels, and the performance improvement will be even higher as the number of channels becomes smaller from our previous simulation results.

E. Effect of the average channel availability

In Fig. 10 and Fig. 11, we study how the average channel availability impacts the performances of the sensing orders. Our proposed scheme is seen to outperform other schemes when the average channel availability changes.

It is interesting to see that as the average availability of the channels becomes larger, the normalized throughput of each user increases while the collision probability also increases. This can be explained as follows. If the average channel availability is higher, multiple users are more likely to find an channel idle, and a later user has to sense another channel occupied by SU that takes the channel first, thus increasing the collision probability. However, when the channels are more likely to be available, the total number of available channels to use increases thus leading to a higher normalized throughout. On the other hand, with a smaller number of channels to use, it is more critical to increase the sensing efficiency and reduce the overhead. Therefore, our proposed scheme has a higher performance improvement when the channel availability is lower.

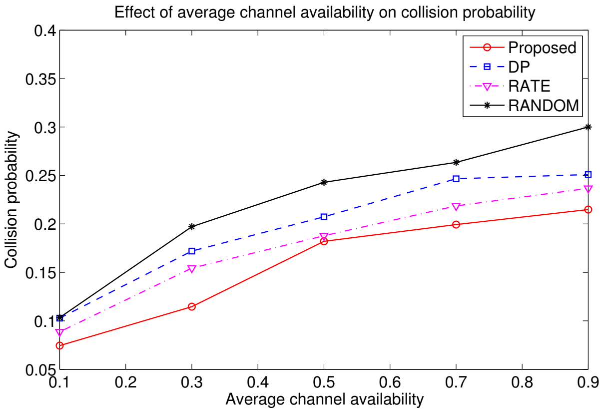

Fig. 11. Effect of average channel availability on collision probability.

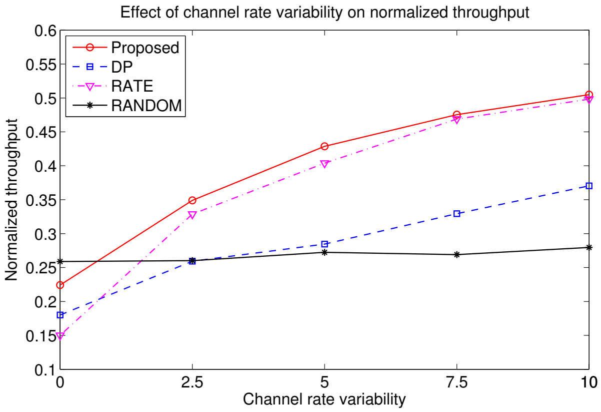

Fig. 12. Effect of channel rate variability on normalized throughput.

F. Effect of the channel rate variability

In Fig. 12 and 13 we study the impact of channel difference among users on the performance, and the difference is represented as the channel rate variability. From the figures we can see that for RANDOM, the normalized throughput and collision probability don't change much because the random selection order does not depend on the channel rate variability.

When the variability equals zero, i.e. the maximum achievable rate of a channel is the same (or very similar) for every user, RANDOM sensing order can outperform other sensing orders including the proposed one. In this case, it is likely that in the other schemes (including our proposed scheme) the users will get very similar sensing orders, which will result in more collisions and less normalized throughput. However, when the channel rate variability increases, all schemes will gradually outperform RANDOM sensing order.

For all three algorithms, when the channel rate variability increases, the normalized throughput also increases while the collision probability is slightly reduced. In fact, Fig. 12 and Fig. 13 also imply that the channel achievable rates might be a more important fact than the channel availability in the multi-user case. Neighboring users might have very similar perception of channel availabilities, if they only consider availabilities for the sensing order, collisions are very likely to occur. The channel rate variability can introduce difference and variety into the sensing order for each user, which would help reduce collisions among SUs. This also explains why the normalized throughput of RATE can approach that of our proposed sensing order when the channel rate variability is large in Fig. 12. What's more, when the variability ranges from 2.5 to 10, the throughput improvement percentage of RATE outperforms DP (DP depends on both channel availability and achievable rates but does not consider collisions) increases from 30% to 37%. This also implies in multi-user case, the channel maximum achievable rate might be a dominating factor and have more impact on the sensing order compared to the channel availability.

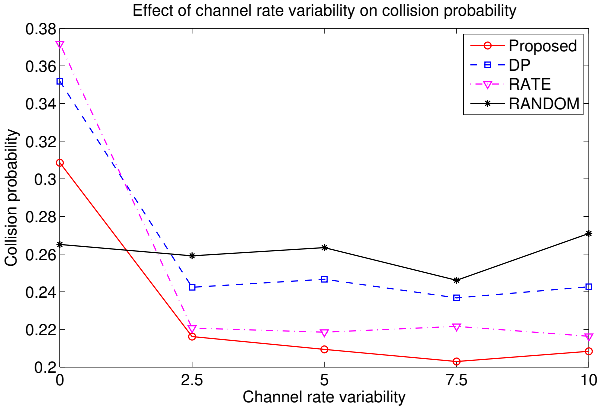

Fig. 13. Effect of channel rate variability on collision probability.

VII. CONCLUSIONS

In this paper we focus on the sensing order problem for multi-user and multi-channel cognitive radio networks. While most of the literature studies discuss the sensing order for a single user, we investigate the scenario in which multiple secondary users sequentially sense and access the channel according to their own sensing orders. In multi-user SU network, collisions among secondary users will lead to performance degradation. We propose a novel methodology that concurrently consider the channel availability, transmission rate and collision probability, and an efficient dynamic programming algorithm to establish the sensing order based on the metric to improve the sensing efficiency and transmission throughput. Simulation results demonstrate that our algorithm can not only efficiently reduce the collisions among secondary users, but also can achieve higher network throughput than other schemes. Furthermore, we also discuss how the network environment affects the performance of our proposed metric.

REFERENCES

[1] Federal Communications Commission, "Spectrum policy task force report," 2002.
[2] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol.23, no.2, pp.201-220, Feb. 2005.

[3] Q. Zhao and B. M. Sadler, “A survey of dynamic spectrum access,” *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 79-89, May 2007.

[4] A. Goldsmith, S. A. Jafar, I. Maric and S. Srinivasa, “Breaking Spectrum Gridlock With Cognitive Radios: An Information Theoretic Perspective,” *Proceedings of the IEEE*, vol.97, no.5, pp.894-914, May 2009.

[5] N. B. Chang and M. Liu, “Optimal channel probing and transmission scheduling for opportunistic spectrum access,” in *Proc. 13th ACM Annual International Conf. Mobile Computing Networking (MobiCom ’07)*, pp.27-38, Montreal, QC, Canada, Sept. 2007.

[6] H. Jiang, L. Lai, R. Fan, H.V. Poor, “Optimal selection of channel sensing order in cognitive radio,” *IEEE Transactions on Wireless Communications*, vol.8, no.1, pp.297-307, Jan. 2009.

[7] H. T. Cheng and W. Zhuang, “Simple Channel Sensing Order in Cognitive Radio Networks,” *IEEE Journal on Selected Areas in Communications*, vol.29, no.4, pp.676-688, April 2011.

[8] R. Fan and H. Jiang, “Channel Sensing-Order Setting in Cognitive Radio Networks: A Two-User Case,” *IEEE Transactions on Vehicular Technology*, vol.58, no.9, pp.4997-5008, Nov. 2009

[9] T. Shu and M. Krunz, “Throughput-efficient sequential channel sensing and probing in cognitive radio networks under sensing errors,” in *Proc. 15th ACM Annual International Conf. Mobile Computing Networking (MobiCom ’09)*, pp.37-48, New York, NY, USA, 2009.

[10] H. Kim, K. G. Shin, “Fast Discovery of Spectrum Opportunities in Cognitive Radio Networks,” in *3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN 2008)*, pp.1-12, Oct. 2008.

[11] S. Yin, D. Chen, Q. Zhang, M. Liu, Shufang Li, “Mining Spectrum Usage Data: A Large-Scale Spectrum Measurement Study,” *IEEE Transactions on Mobile Computing*, vol.11, no. 6, pp. 1033-1046, June 2012.

[12] K. Tsagkaris, A. Katidiotis, P. Demestichas, “Neural network-based learning schemes for cognitive radio systems,” *Computer Communications*, vol.31, no.14, pp.3394-3404, Sep. 2008.