Fungible Orthogonal Channel Sets for Multi-user Exploitation of Spectrum

Irene Macaluso*, Hamed Ahmadi*, Luiz A. DaSilva *

*CTVR Telecommunications Research Center, Trinity College Dublin, Ireland
†Department of Electrical and Computer Engineering, Virginia Tech, U.S.A.

Abstract—This paper proposes a two-stage process for assigning fungible orthogonal channel sets to multiple cognitive radios (CRs) for opportunistic spectrum access. Assigning orthogonal channel sets to the CRs eliminates the possibility of collision among them, and allows the CRs to focus on avoiding collisions with the primary user (PU). In particular, each CR uses a learning-based dynamic channel selection (DCS) algorithm to maximize the exploitation of the assigned channels. We propose a neural network that can accurately estimate the performance of the adopted learning-based DCS algorithm on a set of channels, using the duty cycle and the complexity of the PU’s behavior on the channels. Our simulations on synthetic and real measurement data sets show that the proposed channel sets allocation algorithm, together with the neural network, significantly outperforms a method that selects channels with the lowest duty cycle.

I. INTRODUCTION

As a response to the increasing demand for spectrum, steps are being taken to define more sustainable models of spectrum resources utilization. In Europe the Electronic Communications Committee is currently defining License Shared Access (LSA) [1], a regulatory framework that targets underutilized spectrum bands below 6 GHz. Following the release of the US President’s Council of Advisors on Science and Technology (PCAST) report [2], the Federal Communications Commission has started consultation activities in 3.5 GHz bands for shared access. The idea behind these two frameworks is to allow a set of entities access to spectrum whenever and wherever it is unused by the incumbent users. In particular, the PCAST model envisages two types of shared access: 1 Secondary access users will be granted short term priority operating rights for a given frequency in a specific geographic location while being protected from interference caused by opportunistic access. 2 General authorized access users will be allowed opportunistic access to spectrum that is not used by incumbents and secondary access users. In the terminology of dynamic spectrum access (DSA) both incumbents and secondary access users are primary users (PUs) from a general authorized access user’s point of view. Opportunistic channel access will be enabled by the use of geolocation databases and/or cognitive radio (CR) capabilities.

The model presented in this paper rests on the general authorized spectrum access defined in [2]. In particular, we consider general authorized access users with CR capabilities. In our model each CR network is issued opportunistic operating rights on a group of channels (channel set allocation stage) to be later exploited using a learning-enhanced dynamic channel selection (DCS) approach (DCS stage). Different CRs are assigned orthogonal channel subsets from the set of available channels. This two-stage process increases the probability that an opportunistic network will be able to operate without interruption. In this system each CR proactively selects the channel that is most likely to be available in the future from the given channel subset. The DCS stage is independently implemented by each opportunistic network. The channel set allocation is performed centrally; this stage could be implemented in the Radio Access Coordination and Management and Optimization module envisioned in [2].

The channel sets allocation is in essence a spectrum assignment problem in which the allocation of resources to a CR is based on the performance of the DCS algorithm that will eventually be used by the CR to opportunistically exploit the channels. Hence, a fundamental requirement of the proposed approach is the ability to predict the performance of the adopted DCS approach in correspondence to different channel sets. In order to be able to predict the DCS performance, it is imperative to understand which factors affect it.

Previous works on learning-based DCS investigate the performance of the proposed learning algorithms with respect to the duty cycle (DC) of the channels. However, our studies show that the unpredictability associated with the PU’s wireless resource usage is also a crucial aspect to be taken into account [3], [4]. In [3], we quantified the predictability of the PU activity using the normalized Lempel-Ziv (LZ) complexity and we proved that it is a theoretically sound and practical metric for predicting the effectiveness of reinforcement learning for DCS. In [4], we empirically demonstrated that the LZ complexity is instrumental in the analysis of the effectiveness of the Markov process-based learning algorithm adopted in this paper. In this paper we present a theoretical proof of the relationship between the effectiveness of the Markov process-based learning approach and the complexity of the spectrum occupancy sequences. Moreover, while our work in [4] focused on the empirical analysis of the relationship between the effectiveness of Markov process-based learning algorithm and the DC and LZ complexity of the observed channels, the channel subset allocation proposed in the current paper makes use of these two features in a proactive way to assign to each CR the most effective subset of channels. In fact, our results show that it is possible to accurately predict the performance.
of the Markov process-based learning DCS algorithm using the DC and LZ complexity to characterize the activity of the observed channels.

An interesting consequence of the proposed approach concerns the number of channels assigned to each CR. In practice, under the assumption of orthogonal channel subsets, the number of channels allocated to each CR is limited by the number of CRs and the number of channels available for secondary exploitation in the system. Generally, limiting the number of channels, each CR can exploit is advisable in DCS approaches to preserve computational resources. It also reduces the delay associated with channel sensing. Our results show that this is not a limiting factor of the proposed approach. In fact, our theoretical analysis of the properties of the Markov process-based learning algorithm shows that the performance improvement that a CR can achieve by increasing the number of sensed channels decreases exponentially to zero. This theoretical finding is validated by our empirical analysis of the proposed channel subset allocation, which relies on actual spectrum measurement data.

The allocation of orthogonal channel sets to CRs can be recast in terms of spectrum fungibility [5]. The term fungibility indicates “the property of a good or a commodity whereby individual units are capable of mutual substitution”. For example, since one ounce of gold is equivalent to any other ounce of gold, gold is perfectly fungible. The fungibility of spectrum is commonly assumed in market-based spectrum management [5], [6]. It is also implicitly assumed in most works on secondary spectrum access in that channels are generally considered as fully interchangeable. In [5] the authors discuss a number of factors that influence spectrum fungibility. Among them there exists a temporal dimension that is related to the availability of the channel and is especially significant in the context of opportunistic spectrum access (OSA). In allocating channel sets to multiple CRs, we extend the concept of fungibility to a group of channels. In our work we introduce a new dimension, namely the LZ complexity of PU activity, to the list of dimensions that characterize a channel for secondary user exploitation.

In this paper calligraphic letters (e.g., N) represent sets, E[.] denotes the expected value of a random variable, and |.| denotes the cardinality of a set. The remainder of the paper is organized as follows. Section II contextualizes our work within the current state-of-the-art in opportunistic spectrum access. Section III describes the problem of channel sets allocation that we consider in this paper. Section IV outlines the Markov process-based learning algorithm and presents the theoretical analysis of the relationship between its performance, the number of channels the CR tries to exploit and the predictability in the channel usage. Section V discusses the use of feed-forward neural networks to predict the performance of the DCS algorithm given the duty cycle and LZ complexity values of the set of observed channels. Section VI and Section VII present the results of the proposed channel set allocation approach in the case of single and multiple CRs respectively. We summarize our conclusions in Section VIII.

II. STATE OF THE ART

We can identify two broad classes of approaches in sensing-based OSA: (i) CRs only exploit the average level of activity in each of the channels [7]–[11]; (ii) CRs try to learn the behavior of the PU and to predict the availability of each channel on the next time slot [12]–[14].

The works falling into the first class generally use the duty cycle of the channels as the primary factor in the channel selection mechanism. Both single-user and multi-user scenarios are considered. In some of these works the CR can sequentially sense more than one channel in a time slot and access the free one(s) [10], [11]. The goal is to optimize the sensing order to minimize the number of channels the CR needs to sense before finding an idle channel. In other models CRs are allowed to sense only one channel per time slot [7]–[9]. In the first case, the underlying assumption is that the time slot duration is long enough to allow the CRs to sense the channels. Hence, the PU activity is assumed to be slowly changing if the number of channels the CRs try to exploit is large. In the latter case the problem is often formulated as a multi-armed bandit, either in a fully distributed form [7], [8] or in a centralized setup [9]. The sensing requirements are light in that a CR is only allowed to sense one channel per time slot. The price of this simplicity is that the maximum expected fraction of the time that the CR is able to transmit is $1 - DC$, for the channel with the lowest DC.

Our approach takes a different direction in that we take advantage of the time correlation in the PU activity on a channel so that a CR is able to select the channel with the highest probability of being free in the next time slot. This way we overcome the limitation on the maximum expected fraction of the time during which the CR is able to transmit. Moreover, since our theoretical and empirical results show that the number of channels that a CR can observe does not significantly affect its performance, we can relax the requirement on the duration of steady PU activity, even though our approach assumes full observability of the channels’ state.

Approaches belonging to the second class aim to exploit the regularities in the PU activity to select the channel with the highest probability of being free in the next time slot. Usually full-observability of the channel state is assumed; a variety of methods - from neural network [13] to hidden Markov model [14], Q-learning [15] - have been proposed. To deal with partial observability some authors have proposed the adoption of a partially observable Markov decision process [12]. To the best of our knowledge, all the works in the second class have only addressed the single-user case.

Our approach, while belonging to the second class, considers a multi-user scenario by introducing a spectrum access system that issues operating rights on a group of channels to each CR. The use of a coordination mechanism to sub-license spectrum resources is envisioned in a variety of proposals [2], [16]. The sub-licensing system proposed in this work allocates to each CR a separate group of channels, which the CRs then try to exploit using the Markov-based learning approach we proposed in [4]. The spectrum access system takes into account the channels’ characteristics to maximize the expected
transmission rate that each CR can achieve.

With respect to our work [4] on the single-user scenario this study makes two significant contributions. First, we present a formal proof of the relationship between the effectiveness of the Markov process-based learning approach and the complexity of the spectrum occupancy learning sequences. Second, we formally prove that the performance improvement that a CR can achieve by increasing the number of sensed channels can have a detrimental effect on the DCS performance. Both contributions have major impact on the channel set allocation stage. In fact, the idea of assigning opportunistic operating rights on a group of channels to CRs relies on the ability to assess the performance of the adopted DCS approach in correspondence to different channel sets. Moreover, our results on the impact of the number of channels exploited by each CR allow us to allocate orthogonal subsets of channels without limiting the number of CRs in the system.

III. SYSTEM

Figure 1 depicts the proposed two-stage process for multi-user DCS. Different channels can have different features, i.e. DC and LZ complexity. This is represented in the figure by using different colors. The spectrum access system evaluates the DCS performance in correspondence to the available channels and allocates disjoint sets of channels to the CRs, which in turn opportunistically exploit them by using the Markov process-based learning algorithm.

\[
\max \min_{C_i, i \in \mathcal{N}} \{u(C_i)\} \\
\text{s.t.} \\
C_i \cap C_j = \emptyset \quad \forall i, j \in \mathcal{N}, i \neq j \\
C_i \in \mathcal{P}_\alpha(S) \quad \forall i \in \mathcal{N},
\]

where \(u(C_i)\) denotes the performance of the DCS approach for the i-th CR in correspondence to the set of channels \(C_i\). The optimization considers all possible combinations of \(N\) orthogonal \(C\) in set \(\mathcal{P}_\alpha(S)\) to maximize the minimum value of \(u(C_i)\) for each user \(i\). We used maxmin to balance the DCS performance among all CRs.

The channel sets allocation stage as formalized in (1) is feasible only if it is possible to predict the performance of the DCS algorithm corresponding to every subset of channels \(C\), i.e. if the function \(u()\) is known. In previous works we studied the performance of two learning-based DCS approaches with respect to the duty cycle of the channels and the unpredictability associated with the primary user’s wireless resource usage, which we quantitatively characterized using the normalized LZ complexity [3, 4]. The LZ complexity evaluates the rate of production of new patterns in a sequence. It is computed by scanning the sequence and incrementing its value every time a new substring of consecutive symbols is found. The value is normalized via the asymptotic limit \(n/\log_2(n)\), where \(n\) is the length of the sequence [17]. In this paper, we show how we can use these two features to approximate the function \(u()\). In particular, we show that, given the DC and the LZ complexity of a set of channels, the performance of the DCS algorithm can be accurately estimated by using a feed-forward neural network (NN).

The optimal sets of channels can be computed via exhaustive search, i.e. by evaluating \(u()\) corresponding to every subset of \(S\) whose cardinality is equal to \(\alpha\). However, as the dimension of the search space of (1) is \(1/|\mathcal{N}|! \prod_{i=0}^{\alpha-1} \binom{|S|-\alpha}{\alpha}\), we also analyze the impact of a greedy search algorithm on the channel set allocation problem.

It is worth noting that we have so far implicitly assumed the DC and LZ complexity of the channels to be known. These two features can be accurately and efficiently estimated, as discussed in [4].

IV. LEARNING-BASED DYNAMIC CHANNEL SELECTION ALGORITHM

In this section we introduce the DCS algorithm and discuss its performance analytically.

A. Markov process-based DCS

We consider each channel separately and denote the presence and absence of the primary user on that channel with a “1” and a “0”, respectively. We assume that channel observations are performed periodically and the channel sensing is ideal. Therefore, the availability/unavailability of a channel for secondary use can be represented by a binary sequence.

We create a Markov chain (MC) according to the run length of the zeros or ones strings. The transitions between states depend on the length of the string of consecutive zeros or ones.
observed. Positive states represent the number of observed consecutive ones, and non-positive (negative and zero) states represent the number of observed consecutive zeros. Being in a positive (negative) state, the system moves to the next higher (lower) state whenever a busy (free) channel is observed, and it goes back to state zero (one) whenever a free (busy) channel is observed. The system will expand itself on the fly by adding new states as needed [18].

After creating the Markov chain from the training data, we can compute the transition probabilities between states by counting the number of times each particular transition occurs in the training sequence. Let us denote the state of channel in the training sequence. Let us denote the Markov process we can compute the transition probabilities between states by any state first order Markov chain (MC) estimated by the Markov process-based learning algorithm.

Let us denote the Markov process by \( \lambda \), where \( \lambda \) consists of the state transitions matrix and the initial state distribution. If we denote the state of channel \( k \) at time \( t \) by \( x_k(t) \), the probability of observing a zero in the next time slot on channel \( k \) will be \( p(0|x_k(t), \lambda) \). Therefore, we predict the observation of the channel by:

\[
\hat{y}_k(t) = \begin{cases} 0 & p(0|x_k(t), \lambda) \geq p(1|x_k(t), \lambda) \\ 1 & \text{otherwise} \end{cases},
\]

(2)

where \( \hat{y}_k(t) \) indicates the predicted observation for time \( t \).

Having a set of channels \( C \subseteq S \) under consideration, the system will select the channel with the highest probability of being free in the next time slot for sensing and transmission. The prediction will be successful if \( y_k(t + 1) = 0 \) where \( \hat{k} = \arg \max_k p(0|x_k(t), \lambda), \forall k \in C \).

B. Theoretical analysis of the Markov process-based DCS

In this section we conduct a theoretical study of the relationship between the effectiveness of the Markov process-based learning approach and the complexity of the spectrum occupancy sequences.

Let us denote by \( T \) a binary random variable that models the payoff of a successful transmission as follows:

\[
T = \begin{cases} 1 & \text{if the channel selected by the Markov process-based learning is free} \\ 0 & \text{otherwise} \end{cases}
\]

(3)

For the purpose of this analysis, we modeled the activity on channels as independent random variables. This is the most popular way of modeling the activity on the channels in literature [11], [13], [14]. Moreover, the analysis conducted by Ghosh et al. in [19] on the same measurement data sets we used for our experiments has shown that the channel idleness probabilities of two adjacent channels are statistically independent. We assume each channel to be represented by the realization of a 2-state first order Markov chain (MC) shown in Figure 2a. Denote the transition probability matrix of this MC by \( (p_{00}, p_{01}, 1-p_{00}) \). It is worth noting that this MC models the PU behavior on each channel, i.e. the binary spectrum occupancy of each channel, whereas the Markov chain discussed in the previous section and used by the DCS algorithm models the transition between runs of consecutive zeros and ones in the channel activity. Figure 2 depicts the two MCs. If the channel activity follows a Markov chain, the MC estimated by the Markov process-based learning algorithm has only 2 parameters. However, this is not the case for a generic channel model activity. In fact, the DCS algorithm described in the previous section is able to capture more complex dependencies on the binary spectrum occupancy.

Before proceeding, a few further facts are needed. First, we observe that for an ergodic source the LZ complexity equals the entropy rate of the source [17]. Hence, in the remainder of this section we analyze the DCS performance with respect to the entropy rate of the MC that models the PU activity. Second, it is important to note that there are infinitely many transition probability matrices that lead to the same stationary distribution. In Theorem 1 we show that the DCS performance on channels with the same stationary distribution, i.e. with the same DC, is minimum when then entropy rate of the MCs is maximum.

**Theorem 1.** Consider \( \alpha \) channels, each the realization of a 2-state first order MC. For any given stationary distribution \( \delta = [\delta_0, \delta_1] \) of the MCs, the Markov-process based learning applied to the \( \alpha \) channels achieves a global minimum expected payoff, when the entropy rate of the MCs is maximum.

The proof can be found in the Appendix. The results in the remainder of this section analyze the impact the number \( \alpha \) of channels that the cognitive radio observes, has on the performance of the Markov process-based learning. We also analyze the relationship between \( \alpha \) and the entropy rate. The proofs can be found in the Appendix.

**Theorem 2.** Consider \( \alpha \) channels, each the realization of a 2-state first order MC. For any given stationary distribution \( \delta = [\delta_0, \delta_1] \) of the MCs such that \( \delta_0 \neq p_{00} \), the expected payoff of the Markov-process based learning applied to the \( \alpha \) channels increases with \( \alpha \) and its increment decreases exponentially with \( \alpha \). If \( \delta_0 = p_{00} \), the increment of the expected payoff is null.

It is worth noting that the policy learned by the Markov-process based learning algorithm under the conditions formalized in Theorem 1 and 2 is equivalent to the optimal policy learned by the reinforcement learning algorithm we analyzed in [3], when no cost of switching channel is included.

Let us denote by \( E[T'_{T_{\alpha+1}}] \) the expected value of \( T \) resulting from the exploitation of \( \alpha + 1 \) channels, each of them being the realization of a 2-state first order Markov chain \( m' = (p_{00}', 1-p_{00}) \) with stationary distribution \( [\delta_0, \delta_1] \) and entropy rate \( h' \). Moreover, \( E[T''_{T_{\alpha+1}}] \) denotes the expected value of \( T \) resulting from the exploitation of \( \alpha \) channels, each of them being the realization of a 2-state first order Markov chain \( m'' = (p_{00}'', 1-p_{00}'') \) with stationary distribution \( [\delta_1, \delta_1] \) and entropy rate \( h'' \).
algorithm and we discovered that its performance depends not only on the DC of the observed channels, but also on the predictability of the channel usage, which we quantitatively characterize using LZ complexity.

We now turn our attention to using these two features in a proactive way. In particular, we show that a CR can accurately predict \( E[T] \) given the DC and the LZ complexity values of a set of observed channels.

Predicting \( E[T] \) given the DC and the LZ complexity values of a set of observed channels can be cast as a regression problem, which we address by using a feed forward neural network with a single layer of hidden units. In the remainder of this section we assume that a CR can observe a subset of at most \( \alpha \) channels and can estimate the corresponding LZ and DC values. We trained \( \alpha - 1 \) different neural networks, one for each value of the cardinality of the channel subset under consideration\(^1\). The inputs to each network are the DC and LZ values of the observed channels; the output is the corresponding \( E[T] \). Hence, the number of inputs to each network is \( 2 \times |C| \), with \( |C| \in \{2, 3, \ldots, \alpha\} \).

We tested the accuracy of the proposed approach relying on both an idealized mathematical model of PU behavior and on actual PU activity data. In the first case, we generate synthetic data for PU activity according to a two-state Markov chain. We vary the parameters of the Markov model to experiment with different levels of PU activity, characterized by the probability that a PU is active in a channel, and with different levels of complexity in the pattern of activity, characterized by the LZ complexity. PU activity in each channel is assumed independent.

In the latter case, we rely on spectrum measurements conducted at RWTH Aachen University to determine, for any given time slot and set of channels, whether there is PU activity. Again the channels are of same bandwidth, but this time no other assumptions are made regarding the activity in each channel, and the solution is evaluated using real sensing data.

A. Neural Network with Synthetic Data

In this case we trained the neural networks over datasets generated by modeling the channels as independent random variables. Each channel is the realization of a 2–state first order MC. Therefore we generated MCs with different values of stationary distribution and LZ complexity. We considered 7 possible \( \delta_0 \) values in the range \( 0.2, \ldots, 0.8 \). For each of these values, we considered \( p_{11} \) (or \( p_{00} \)) values of 0.1, 0.3, 0.5, 0.7, 0.9 if \( \delta_0 >= 0.5 \) (if \( \delta_0 < 0.5 \)), obtaining 35 different transition probability matrices. This is the set \( S \) of channels that we used to train the neural networks.

In particular, for each value of the cardinality of the channel subset \( C \) under consideration, we generated the corresponding training set by evaluating the frequency of success of the learning algorithm for each of the \( \binom{35}{C} \) possible combinations. Details about the number of samples used to train and evaluate the Markov process-based learning DCS algorithm can be found in Section VI. We trained each neural network over the corresponding training set using the Levenberg-Marquardt algorithm [20].

In order to test the prediction accuracy of the networks, we generated additional MCs corresponding to stationary distributions not included in the datasets described above. We considered 7 additional possible \( \delta_0 \) values in the range 0.15, \ldots, 0.75. For each of these values, we considered \( p_{11} \) (or \( p_{00} \)) values of 0.1, 0.3, 0.5, 0.7, 0.9 if \( \delta_0 >= 0.5 \) (if \( \delta_0 < 0.5 \)), obtaining 35 different transition probability matrices. This is the set \( S' \) of channels that we used to test the neural networks. Again, for each value of the cardinality of the channel subset \( C \) under consideration, i.e. for each network, we generated the corresponding test set by evaluating the frequency of success of the learning algorithm for each of the \( \binom{58}{C} \) possible combinations. Table I shows, for each neural network, the proportion of samples for which the absolute error is less than 0.03 and 0.05 for the training set and testing set described above. Since our theoretical analysis of the properties of the Markov process-based learning shows that the performance improvement that a CR can achieve by increasing the number \( \alpha \) of sensed channels decreases exponentially with \( \alpha \) (see Theorem 2 in Section IV-B), we consider a CR capable of observing a subset of at most \( \alpha = 5 \) channels.

It can be observed that the prediction accuracy of each neural network is extremely high in both the training and testing set. This result shows that each neural network is able to accurately predict the corresponding \( E[T] \) even when presented with channels exhibiting new values of stationary distributions and complexity. Hence, a CR can reliably use the output of the neural networks as accurate estimates of function \( u() \) in (1).

B. Neural Network with Measurement Data

In order to analyze for real spectrum data the accuracy of feed-forward neural networks, we relied on spectrum measurements conducted at RWTH Aachen University [21], in

---

\(^1\)It should be noted that a CR needs to observe at least 2 channels in order to use the Markov process-based learning.
frequency bands ranging from 20 MHz to 6 GHz.

To build the set \( S \) of all the channels used to train the neural networks, we considered sequences of spectrum occupancy over 12 hours (from 11:00 to 23:00) in four frequency bands: the 2.4 GHz ISM band, the DECT band, and the GSM900 and GSM1800 bands. For each value of the cardinality of the channel subset \( C \) under consideration and for each band, we evaluated the \( E[T] \) of the Markov process-based learning DCS algorithm for all the possible combinations of \(|C|\) channels with duty cycle \( DC \in [0.3, 0.8] \). The training set of the neural network corresponding to \(|C|\) under consideration is the union of all the above combinations.

We generated the test set for each network using the same procedure and considering sequences of spectrum occupancy over 12 hours on a different day. Table II shows, for each neural network, the proportion of samples for which the absolute error is less than 0.03 and 0.05 for the training set and testing set described above. Again, we consider a CR can observe a subset of at most \( \alpha = 5 \) channels.

### VI. SINGLE-USER CHANNEL SUBSET SELECTION

In this section we analyze the impact of the proposed approach for a single-CR scenario. In this case the channel sets allocation problem formulated in (1) simplifies to the search for the subset of channels of cardinality \( \alpha \) that corresponds to the optimal performance of the DCS algorithm:

\[
C^* = \underset{C \in \mathcal{P}_\alpha(S)}{\arg \max} u(C) \tag{4}
\]

In Section V we showed that it is possible to accurately estimate the performance function \( u() \) given the DC and LZ complexity of the set of observed channels, by using feed-forward neural networks. Having approximated \( u() \), we can solve the problem in (4) by exhaustive search, i.e. evaluating the performance function for every subset of \( S \) whose cardinality is equal to \( \alpha \).

In order to reduce the search space, we investigated the impact of the greedy search method on the proposed problem. The greedy algorithm initially selects a channel \( s \) in \( S \) which maximizes the value of \( u(\{s\}) \). The algorithm removes \( s \) from \( S \) and adds it to \( D^* \), where \( D^* \) is the set of selected channels by the greedy algorithm. In the next steps, the algorithm again selects channel \( s \) from \( S \) that maximizes \( u(D^* \cup \{s\}) \). Similarly to the first step, the greedy algorithm removes \( s \) from \( S \) and adds it to \( D^* \). This procedure continues until the cardinality of \( D^* \) reaches \( \alpha \), \(|D^*| = \alpha \). This procedure is summarized in Algorithm 1.

In the remainder of this section we present the results of our proposed approach. For all the simulations in this paper, we trained the Markov process-based learning algorithm over \( 10^5 \) samples and evaluated its performance over 24000 unseen samples. Here, each sample represents the presence/absence of the PU on a given channel, at a given time. The number of samples used for training and evaluation is the same for synthetic data and real spectrum measurements. In the case of real spectrum measurements we considered sequences of spectrum occupancy over 12 hours (from 11:00 to 23:00). We retrain the algorithm every \( 5 \times 10^3 \) samples by the \( 10^3 \) most recently visited samples. Given a set of channels, we estimate \( E[T] \) by the ratio of successful transmissions to total number of time slots. In the rest of this work \( E[T] \) refers to the experimentally estimated \( E[T] \).

For the purpose of understanding the impact of DC and LZ complexity on the channel subset selection problem, Fig. 3 shows the \( E[T] \) corresponding to different subset selections. We start with the \( E[T] \) resulting from the exploitation of 2 channels, both exhibiting a DC of 0.55. Then, we consider two different options. In one case (star markers in the figure), we add additional channels with DC = 0.6 and low LZ complexity. In the other case (circle markers in the figure), the additional channels have DC = 0.56 and high LZ complexity. The figure shows that relying only on the channels’ DC to select which channels to observe can result in a significant loss of \( E[T] \).

| \(|C|\) = 2 | \(|C|\) = 3 | \(|C|\) = 4 | \(|C|\) = 5 |
|---|---|---|---|
| \(|e| < 0.03\) | \(|e| < 0.05\) | \(|e| < 0.03\) | \(|e| < 0.05\) |
| Training | 0.983 | 1.0 | 0.978 | 1.0 |
| Testing | 0.936 | 1.0 | 0.96 | 1.0 |

Algorithm 1 Proposed channel subset selection greedy algorithm

\[
D^* = \emptyset \quad \text{while } |D^*| \leq \alpha \text{ do} \\
\quad D^* = D^* \cup \{\arg \max_s u(D^* \cup \{s\})\} \\
\quad \text{Remove } s \text{ from } S \\
\text{end while}
\]

Fig. 3: Comparison of the probability of successful transmission for different compositions of the observed channel set. In the case of 2 channels, \( E[T] \) refers to channels with DC= 0.55. Star markers correspond to \( E[T] \) when we add channels with DC= 0.6 and low LZ complexity. Circle markers correspond to \( E[T] \) when we add channels with DC= 0.56 and high LZ complexity.
We now turn to looking at the performance of the greedy algorithm when the performance function $u()$ is known. Initially we create a set of 12 two-state MCs that can model channels with three different DC (DC $\in \{0.55, 0.57, 0.6\}$) and four different LZ complexity values for each DC, following the method explained in Section V-A. In Fig. 4, we compare the Markov process-based learning algorithm performance on the optimum channel subset, on the subset of channels with lowest DC, and on the subset selected by the greedy algorithm. The optimum channel subset, i.e., the solution to (4), is computed by exhaustive search, i.e., evaluating the performance of the Markov process-based learning algorithm on all the possible subsets of channels and selecting the subset corresponding to the highest performance. This provides an upper-bound for the performance of any channel selection algorithm. Selection of the channel subset is also possible by ignoring the LZ complexity and simply selecting the channels with lowest DC. Since there exist multiple channels with the same DC in the set of channels under consideration, we compute $E[T]$ considering all the possible combinations of channels with lowest DC from the set. For example, when $\alpha = 3$ Fig. 4 shows the average $E[T]$ of all the 4 possible combinations of channels with $DC = 0.55$. The figure illustrates that the proposed greedy algorithm selects the optimum subset of channels in the case of synthetic data. We also measured the performance of the Markov process-based learning algorithm observing all 12 channels. The corresponding $E[T]$ is 0.851, which is only slightly larger than the $E[T]$ obtained when observing only 5 channels.

The results so far presented assume that the function $u()$ is known. In Section V, we introduced feed forward neural networks to estimate $E[T]$ given the DC and LZ complexity of the observed channels ($C$). Moreover, we presented its accuracy in estimating $E[T]$ over the synthetic and real measurement data in Table I and Table II, respectively. We now turn to looking at the impact of using the estimates provided by the neural networks into the channel subset selection problem. The results in Fig. 6 show the difference (error) between the $E[T]$ corresponding to the optimum subset and the $E[T]$ corresponding to subset selected using the estimates provided by the neural networks to solve (4) for all channels in the ISM band ($DC \in [0.3, 0.8]$). In particular, $e_1$ corresponds to the loss in performance when we use exhaustive search; and $e_2$ denotes the error when using the greedy algorithm. The figure shows this difference in performance for both training and test sets described in Section V.

These results show that the proposed approach is able to select a near-optimal subset of channels. In fact, the loss in performance with respect to the optimum channel subset is at most 0.06. Moreover, using the greedy algorithm has a very

| $|C| = 2$ | $|C| = 3$ | $|C| = 4$ | $|C| = 5$ |
|-----------------|-----------------|-----------------|-----------------|
| $|C| = 2$ | $|C| = 3$ | $|C| = 4$ | $|C| = 5$ |
| Training | 0.91 | 0.97 | 0.93 | 0.99 | 0.94 | 0.95 | 0.95 |
| Testing | 0.82 | 0.92 | 0.92 | 0.94 | 0.88 | 0.95 | 0.95 |

**TABLE II:** Measurement data: proportion of samples corresponding to absolute error values less than 0.03 and 0.05. The error is computed as the difference between the frequency of success evaluated in simulations and the output of the neural network.

![Fig. 4: Synthetic data: Performance comparison of greedy algorithm, optimum selection resulted by exhaustive search and simply selecting the channels with lowest DC.](image1)

![Fig. 5: Performance comparison of greedy algorithm, optimum selection resulted by exhaustive search and simply selecting the channels with lowest DC. Real measurement data for the 2.4GHz ISM band.](image2)
limited impact on the results and can sometimes outperform the NN-based exhaustive search solution. It should be noted that this is possible because the search for the optimum channel subset is performed using an approximation of $u(\cdot)$ provided by the neural networks.

VII. SELECTING FUNGIBLE CHANNEL SETS FOR MULTIPLE USERS

In previous section we proposed a greedy algorithm for channel subset selection problem and evaluated its performance over synthetic and real measurement data. This algorithm provides a greedy solution for the single-user scenario, which is the special case of (1). In this section we generalize the scenario that we studied Section VI by considering multiple CRs in the network.

Similar to the previous section, we propose a greedy algorithm to reduce the search space of (1). To assign fungible disjoint subsets of channels to the CRs, the proposed greedy algorithm begins by creating a list of the CRs which are randomly ordered. Let us denote this list by $n$ where $n(i) \in N$, $i \in \{1, \ldots, N\}$ is the $i^{th}$ element of the list. Then the algorithm makes a matrix $A$ where $a_{s_i, m} = u(s_i, m)$ for all possible $s_i, m \in S, \forall l, m \in \{1, \ldots, S\}$ is computed and saved in the corresponding element of the matrix $a_{l,m}$. For each CR in the list $n$ (list of CRs) the algorithm assigns the channel $s_i$ to the CR if the sum of the elements in row $l$ of $A$ has the maximum value. The algorithm removes channel $s_i$ from $S$ (the set of available channels) and adds it to $D_{n(i)}$, where $D_{n(i)}$ is the set of selected channels for user $n(i)$ by the greedy algorithm. The greedy algorithm also removes the corresponding row and column of the selected channel from $A$. The algorithm repeats this procedure for the next CRs until it gives each CR a channel.

In the next steps, the greedy algorithm reverses the order of users (reverses elements of $n$); starting from the first user in the list, the greedy algorithm selects channel $s$ from $S$ that maximizes $u(D_{n(i)} \cup \{s\})$. Similarly to the first step, the greedy algorithm removes $s$ from $S$ and adds it to $D_{n(i)}$. The algorithm continues until it reaches the end of the list of CRs. The algorithm then reverses the order of users again and continues the aforementioned procedure until the cardinality of $D_{n(i)}$ reaches $\alpha$ for all users ($|D_{n(i)}| = \alpha, \forall i \in N$), or there exists no channels to assign i.e., $S = \emptyset$. This procedure is summarized in Algorithm 2.

Algorithm 2: Proposed multiuser channel subset selection greedy algorithm

\[
\begin{align*}
  &D_{n(i)} = \emptyset, \forall i \in N \\
  &\text{Initialize matrix } A \text{ as } a_{s_i, m} = u(s_i, m), \forall l, m \in \{1, \ldots, S\} \\
  &\text{for } i = 1 \text{ to } |N| \text{ do} \\
  &\quad D_{n(i)} = \arg \max_s \sum_{j=1}^{|S|} a_{s,j} \\
  &\quad \text{Remove } s \text{ from } S' \\
  &\text{end for} \\
  &\text{while } |D_{n(i)}| \leq \alpha, \forall n \in n, \text{ and } S \neq \emptyset \text{ do} \\
  &\quad n = \text{reverse of } n \\
  &\quad \text{for } i = 1 \text{ to } |N| \text{ do} \\
  &\quad \quad S' = \emptyset \\
  &\quad \quad \text{if } S' = \emptyset \text{ then} \\
  &\quad \quad \quad \text{EXIT} \\
  &\quad \quad \text{end if} \\
  &\quad \quad D_{n(i)} = \arg \max_s u(D_{n(i)} \cup \{s\}) \\
  &\quad \quad \text{Remove } s \text{ from } S' \\
  &\quad \text{end for} \\
  &\text{end while}
\end{align*}
\]

Algorithm 2 results in disjoint fungible subsets of channels. We measure the performance of Algorithm 2 when $u(\cdot)$ is estimated by the proposed neural network. We benchmark these results against the results of subset selection when $u(\cdot)$ is computed by the Markov process-based algorithm and the subsets are selected as solution of (1) using CPLEX [22]. We call the former neural network-based greedy (NNG) and the latter optimum. We also compare the performance of the proposed solution with the performance obtained by assigning to the CRs the subsets of channels with Lowest DC (LDC).

It is worth noting that all three approaches (optimum, NNG and LDC) require a centralized coordination mechanism. While the complexity to compute the optimum is too high for any realistic scenario, the computational requirements for NNG and LDC are comparable and modest. Both solutions require an initialization step. In the case of LDC, the initialization step consists of sorting the channels according to their DC values; hence its complexity is $O(|S| \log |S|)$. The
initialization of matrix $A$ in Algorithm 2 considers all the possible $2^{|S|}$-combinations of the $|S|$ channels and for each of them the estimation of $u(i)$ is performed by a feedforward neural network, whose complexity is linear in the number of inputs to the network (in this case case 4). Overall the initialization step complexity is $O(|S|^2)$. Algorithm 2’s main cycle is executed $\alpha \times N$ times; for each iteration the complexity to estimate $u(i)$ is linear in the number of inputs to the neural network (in this case at most $2\alpha$). Finally, the main LDC module’s complexity is $O(\alpha N)$. As we will show in the remainder of this section, the higher complexity of the NNG approach results in a significantly better performance for the CRs.

We consider the scenario that there exists three CRs in the network and there are 20 channels which they can access. In Fig. 7 and Fig. 8, 10 channels have high LZ complexity ($\sim 0.9$) and the other 10 channels have low LZ complexity ($\sim 0.2$). The channels with high complexity have lower DC (we will call it base-DC) and the channels with low LZ have higher DC. The difference between the DCs of the two groups of channels is denoted by $\Delta$. In Fig. 7 we present the average difference between the performance of the Markov process-based learning algorithm on the optimally selected subsets and the subsets assigned using LDC. We considered 3 different base-DC values (0.35, 0.45 and, 0.55) and increased $\Delta$. Fig. 7 shows that the difference between the performance of subsets selected by the two methods is higher when $\Delta$ is smaller and the difference reduces as $\Delta$ increases. Moreover, this difference is larger when the base-DC is larger. Fig. 8 presents a comparison between the performance of LDC, NNG and the optimum when the base-DC is 0.35. As expected in this figure we see that the performance of the LDC method does not change with respect to the increase in $\Delta$ because this method always selects the channels that have the lowest DC (base-DC in this case). More importantly the figure shows that the NNG algorithm performs as well as the optimum which significantly outperforms LDC.

We tighten the difference between the LZs’ of the two groups of channels in our next set of simulations. Additionally, we tested our algorithms on a larger system with 60 channels and 9 CRs. In Fig. 9 and Fig. 10, we consider a scenario where 30 channels out of the 60 channels have base-DC and high LZ values and the other 30 channels have medium LZ values and higher DC (base-DC+\Delta). All CRs are assigned $\alpha = 3$ channels to exploit using the Markov process-based algorithm. Fig. 9 shows the average difference between the performance of Markov process-based algorithm over the channel subsets selected by the NNG algorithm and the channel subsets that have the lowest DC. Fig. 10 shows the comparison of the three methods in this scenario when the base-DC is 0.45.

The figures show that using the NNG method for assigning channels to the CRs results in significantly higher performance of the Markov process-based learning algorithm than assigning the channels with the lowest duty cycle to the CRs. Moreover, the figures illustrate that NNG’s performance is slightly lower than the optimum selection.

We also compared the performance of the proposed multi-user channel subset selection methods over a real measurement
data set. Fig. 11 presents the performance of the Markov process-based learning algorithm when channel subsets are selected by the optimal method, NNG and LDC. We were not able to compute the optimum channel subsets when each subset has 4 channels due to its large search space.

The results are achieved over a data set of all channels in 2.4 GHz ISM band with DC ∈ [0.3, 0.8] over a period of 12 hours. The figure illustrates that the NNG channel selection method outperforms selecting the channels with the lowest DC on real data similar to the results obtained with synthetic data, and its results are close to the optimum. The figure also shows another perhaps more important point. Each CR’s expected performance when using 3 channels is higher than the performance obtained when exploiting a group of 4 channels assigned using LDC. Hence the higher complexity of the NNG allows our solution to reduce the number of channels each CR has to monitor, thus preserving the CRs’ computational resources and reducing the delay associated with channel sensing. This is of particular importance when one considers that the computational resources of each CR are a much more expensive commodity - e.g. in terms of battery life - than the computational resources of a central coordination unit.

VIII. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we proposed a two-stage process for assigning fungible orthogonal channel sets to a group of CR networks for secondary use. In our model opportunistic operating rights on a set of channels are granted to CR networks that use a learning-enhanced DCS algorithm to efficiently exploit the assigned channels. We used the performance of the proposed learning-enhanced DCS algorithm as a fungibility metric to assign orthogonal channel subsets to CRs. This means that the assigned subsets are interchangeable in terms of the DCS performance. We showed that a feedforward neural network can accurately estimate the fungibility of channels sets by using the DC and LZ of the channels. Our simulation results on synthetic and real measurement data showed that using the NN in conjunction with a greedy subset allocation algo-

rithm significantly outperforms the channel selection algorithm based on the lowest DC criterion. Moreover, the performance of NNG is comparable to the optimum allocation.

The proposed two-stage process is an effective way to enable the use of more sophisticated cognitive radio techniques to access spectrum while eliminating the potentially destructive competition among CRs. The pros and cons of a centralized versus distributed mechanism to opportunistically access spectrum are well known. However, when we consider the most recent frameworks currently under development to enable shared access to spectrum, the consensus is to allow secondary access to the spectrum in a coordinated manner. Indeed both LSA and PCAST envision the existence of an access coordination mechanism [1], [2]. Our approach extends this mechanism by allowing each CR to exploit multiple channels, thus increasing its expected transmission rate.

In this work, we assumed that a CR should switch to the channel with the lowest probability of being occupied in the next time slot. The tacit assumption here is that the cost associated with channel switching is negligible. If we consider CR networks that combine the use of proprietary spectrum with opportunistic access to spectrum, the cost of switching channel can be neglected. In this case channel switching could be implemented by using the Activation/Deactivation MAC control element described in the LTE Release 10 [23] for the activation/deactivation mechanism of secondary carrier components. The combined use of exclusive owned spectrum and shared spectrum through carrier aggregation has recently started to attract attention as a way of enabling systematized solutions to flexible spectrum use [24].

When CR networks cannot rely on the exclusive use of a portion of spectrum to facilitate the channel switching, a related cost might be included in the DCS algorithm. In our previous works [3], [15] we considered the effect of channel switching on a reinforcement learning-based DCS algorithm. There the optimal policy depends on the actual cost. In fact, we showed that, for a given DC, when the entropy rate exceeds a threshold, which depends on the cost of switching channel, the radio does not benefit from switching. We intend to investigate the impact of including a cost of switching in the model presented in this paper. In particular, we plan to study how to dimension the cost of switching with respect to the gain of selecting the channel with the highest likelihood of being free in the next time slot.

APPENDIX

A. Theorem 1

Proof. For any given stationary distribution δ = [δ0, δ1], the entropy rate of the MCs with stationary distribution δ has a unique global maximum at p0 = δ0 (proof of Theorem 1 in [3]). If a channel is the realization of a 2-state first order Markov chain, the probability of observing yk(t) = 1 (or yk(t) = 0) does not depend on xk(t) (by definition it only depends on yk(t − 1)). In this case (2) will become

\[ y_k(t) = \begin{cases} 0 & p(0|y_k(t-1), \lambda) \geq p(1|y_k(t-1), \lambda) \\ 1 & \text{otherwise} \end{cases} \]

(5)
\[
\Delta E[T] = E[T_{\alpha+1}] - E[T_{\alpha}] = \begin{cases} 
(1 - \delta_0)^\alpha (p_{00} + p_{11} - 1) & \text{if } p_{00} > \delta_0 \\
\delta_0^\alpha (1 - p_{00} - p_{11}) & \text{if } p_{00} < \delta_0 \\
0 & \text{if } p_{00} = \delta_0 
\end{cases}
\]

\[
E[T_{\alpha+1}'] = \begin{cases} 
\delta_0(p_{00}' - p_{00}) + (1 - \delta_0)\alpha (1 - p_{11}' - 1) & \text{if } p_{00} \geq \delta_0 \\
(1 - \delta_0)\alpha (1 - p_{00}' - p_{11}' - 1) + (1 - \delta_0) (1 - 4\delta_0^{\alpha + 1}) & \text{if } p_{00} < \delta_0
\end{cases}
\]

\[
\Delta E = \begin{cases} 
\delta_0(p_{00}' - p_{00}) + (1 - \delta_0)\alpha (1 - p_{11}' - 1) & \text{if } p_{00} \geq \delta_0 \\
(1 - \delta_0)\alpha (1 - p_{00}' - p_{11}' - 1) + (1 - \delta_0) (1 - 4\delta_0^{\alpha + 1}) & \text{if } p_{00} < \delta_0
\end{cases}
\]

C. Proposition 1

Proof. As \( \delta_0 = 0.5 \), then \( p_{00}' = p_{11}' \) and \( p_{00}'' = p_{11}'' \). Also, \( h(p_{00}) = h(1 - p_{00}) \) for all \( p_{00} \).

Let us consider first the case where \( p_{00}' > \delta_0 \) (i.e. \( p_{00} > 1 - p_{11}' \)). As \( h'' < h' \), for any value of \( p_{00} > \delta_0 \) we need to consider only the values of \( p_{00}'' \in (p_{00}', 1) \) and \( p_{00}'' \in (0, 1 - p_{00}') \). In fact, the entropy rate decreases for any \( p_{00}'' \in \delta_0 \), and increases for any \( p_{00}'' \in (0, \delta_0) \) (see proof of Theorem 1 in [3]). For any value of \( p_{00}'' > p_{00}' \), as both \( m \) and \( m'' \) have the same stationary distribution, it can be verified that \( 1 - p_{11}' > 1 - p_{11}'' \). Hence, when the CR exploits \( \alpha \) channels defined by \( m \) and one channel defined by \( m'' \), the policy learned by the Markov process-based learning can be summarized as follows: if the \( \alpha + 1 \) channels were all busy in the previous time slot, select one channel among the \( \alpha \) channels corresponding to \( m' \); select the channel corresponding to \( m'' \) if it was free on the previous time slot; otherwise select one the channels corresponding to \( m' \) which was free on the previous time slot.

As \( p_{00} < \delta_0 \), it can be verified that \( 1 - p_{11}' < 1 - p_{11}'' \). Also, \( p_{00}' = p_{11}' \) and \( p_{00}'' = p_{11}'' \). Hence, when the CR exploits \( \alpha \) channels defined by \( m \) and one channel defined by \( m'' \), the policy learned by the Markov process-based learning can be summarized as follows: select the channel corresponding to \( m'' \) if it was busy in the previous time slot; if only the channel corresponding to \( m'' \) was free on the previous time slot, select one channel among the \( \alpha \) channels corresponding to \( m' \); otherwise select one of the channels corresponding to \( m' \) which was free on the previous time slot.

Let us now consider the case where \( p_{00} < \delta_0 \) (i.e. \( p_{00} < 1 - p_{11}' \)). As \( h'' < h' \), for any value of \( p_{00} < \delta_0 \) we need to consider only the values of \( p_{00}'' \in (1 - p_{00}', 1) \) and \( p_{00}'' \in (0, p_{00}) \). For any value of \( p_{00}'' < p_{00}' \), it can be verified that \( 1 - p_{11}' < 1 - p_{11}'' \). Accordingly, when the CR exploits \( \alpha \) channels defined by \( m \) and one channel defined by \( m'' \), the policy learned by the Markov process-based learning can be summarized as follows: if the \( \alpha + 1 \) channels were all free in the previous time slot, select one channel among the \( \alpha \) channels corresponding to \( m' \); select the channel corresponding to \( m'' \) if it was busy on the previous time slot; otherwise select one the channels corresponding to \( m' \) which was busy on the previous time slot. For any value

\[\delta_0 = \frac{1 - p_{11}'}{1 - p_{00}'} \]
of $p''_{00} > 1 - p''_{01}$, it can be verified that $p''_{00} > 1 - p'_{11}$ and $p'_{00} > 1 - p'_{11}$. Hence, the policy learned by the CR can be summarized as: select the channel corresponding to $m''$ if it was free on the previous time slot; select one of the channels corresponding to $m'$ which was busy on the previous time slot if the channel corresponding to $m''$ was busy on the previous time slot; otherwise select one of the channels corresponding to $m'$. Then, we can write $E[T'_{\alpha+1}]$ as in (9). After some manipulations of (9), the difference of expected payoffs $\Delta E = E[T'_{\alpha+1}] - E[T'_{\alpha+1}]$ is given in (10).

It can be verified that all equations in (10) are always positive.

**D. Corollary 1**

**Proof.** For any value of $\delta_{0}$, if $p''_{00} > 1 - \delta_{0}$, $E[T'_{\alpha+1}]$ and $\Delta E$ correspond to the first equation in (9) and (10) respectively. If $p''_{00} < 1 - \delta_{0}$, then $E[T'_{\alpha+1}]$ and $\Delta E$ correspond to the third equation in (9) and (10) respectively.

---

**REFERENCES**


