Learning-based Spectrum Sensing in Cognitive Radio Networks via Approximate POMDPs

Bharath Keshavamurthy and Nicolò Michelusi

Abstract

In this paper, an innovative spectrum sensing and access strategy is proposed, wherein a cognitive radio dynamically learns the time-frequency correlation model underlying the occupancy behavior of licensed users in a radio ecosystem via the Baum-Welch algorithm, and concurrently devises – under spectrum sensing constraints and a noisy observation model – an approximately optimal spectrum sensing and access policy (with a normalized sub-optimality gap of 5%) that exploits this learned correlation model via a randomized point-based POMDP value iteration method coupled with fragmentation and Hamming distance state filter heuristics to alleviate the inherent computational complexity; ergo, facilitating regulation of the trade-off between secondary network throughput and incumbent interference. Numerical evaluations demonstrate improvements over state-of-the-art algorithms: 60% over correlation-based clustering, 25% over Neyman-Pearson Detection, 6% over Viterbi, and 7% over adaptive DQNs – with these enhancements more pronounced owing to optimized rate adaptation. The proposed solution is extended to a distributed multi-agent setting with neighbor discovery and channel access rank allocation, which improves throughput by 43% over cooperative TD-SARSA, 84% over cooperative greedy distributed learning, and 3× over non-cooperative learning via g-statistics and ACKs. This multi-agent scheme is implemented on the DARPA Spectrum Collaboration Challenge platform, manifesting superior performance over competitors in a real-world TDWR-UNII WLAN scenario emulation; and its implementation feasibility is validated on an ad-hoc distributed wireless testbed of ESP32 radios, exhibiting 96% success probability.

Index Terms

Hidden Markov Model, Cognitive Radio, Spectrum Sensing, POMDP

Part of this research has been accepted at IEEE ICC 2021 [1].

This research has been funded in part by NSF under grant CNS-1642982.

B. Keshavamurthy is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN.

N. Michelusi is with the School of Electrical, Computer and Energy Engineering, Arizona State University, AZ.

Email: bkeshava@purdue.edu, nicolo.michelusi@asu.edu

I. Introduction

Cognitive radios have been touted as instrumental in solving resource-allocation problems in resource-constrained radio environments. Their adaptive computational intelligence facilitates the dynamic allocation of scarce network resources, particularly the spectrum. With the advent of fifth-generation (5G) cellular technologies [2], [3], a multitudinous array of devices will be brought into the wireless communication ecosystem, resulting in an enormous strain on the available spectrum resources. Dynamic Spectrum Access, the key defining feature of cognitive radio networks, is being widely studied as a solution to the problem of spectrum scarcity, in both military and consumer spheres: cognitive radios intelligently access portions of the spectrum unused by the sparse and infrequent transmissions of licensed users in the network, in order to deliver their own network flows, while adhering to interference compliance requirements. In order to intelligently access the spectrum white-spaces, the cognitive radio – referred to as a Secondary User (SU) – needs to solve for a channel sensing and subsequent access policy based on noisy observations of the occupancy behavior of licensed users or incumbents in the network, referred to as Primary Users (PUs). Yet, critical design limitations, driven by energy efficiency requirements or constraints on sensing times [4], prevent the SU from sensing simultaneously all the channels in the discretized spectrum of interest. Under these constraints, the SU can only sense a small fraction of all available channels, and access among them those that are deemed idle, as studied in [4]–[11]. However, this approach is quite conservative, since it does not allow the SU to access the large pool of channels that have not been sensed.

PU occupancy may exhibit correlation across both time and frequency, as demonstrated in [12] and visualized in Fig. 2. Exploiting this time-frequency correlation structure may significantly improve white-space detection, thus enabling SUs to predict the state of the channels that have not been directly sensed, and unlocking additional opportunities for SU spectrum access. In this paper, focusing first on deployments with multiple PUs and a single SU, we propose to learn the time-frequency correlation statistics underlying the occupancy behavior of PUs in the network via a parametric model, and to concurrently utilize these estimated statistics to construct an approximately optimal sensing and access policy using an inventive Partially Observable Markov Decision Process (POMDP) formulation – solved via randomized point-based value iteration. Furthermore, we extend our single-agent POMDP formulation to distributed and centralized multi-agent deployments enveloping radio environments with several SUs performing intelligent

spectrum sensing and access – with neighbor discovery and channel access rank allocation schemes – to collaborate and capitalize on spectral resources left unused by the multiple PUs in the network: in addition to demonstrating the implementation feasibility of our multi-agent POMDP framework, we also illustrate the performance disparities between collaborative and opportunistic (non-cooperative or competitive) access through comparisons with algorithms in the multi-agent state-of-the-art.

Related Work: Firstly, spectrum sensing and access algorithms in the state-of-the-art have been developed under the assumption that the occupancy behavior of PUs is either correlated across time but independent across frequency [6], [7], or independent across both time and frequency [4], [5], [8]–[11], [13]. These assumptions are not only impractical but also imprudent because critical information aiding the accurate detection of white-spaces can be gleaned by exploiting the correlation in PU time-frequency occupancy behavior; prudently, we exploit both time and frequency correlation. Specifically, [6] outlines a solution for spectrum sensing and access employing Temporal Difference (TD) SARSA with Linear Function Approximation (LFA), with a strictly temporal PU occupancy correlation model – thereby, failing to capitalize on correlations in PU occupancy behavior across frequency.

Secondly, although time-frequency PU occupancy correlation is studied in [7] and [14], they determine the time-frequency PU occupancy correlation structure offline using pre-loaded databases, which is inefficient in non-stationary settings; in contrast, with our solution, SUs learn the parametric model encapsulating PU occupancy correlation via an online Baum-Welch algorithm, and leverage this knowledge to concurrently optimize spectrum sensing and access – under sensing limitations – via approximate point-based POMDP value iteration. The authors in [14] model their solution under a noiseless setting, which is quixotic; instead, we employ an AWGN observation model within our HMM formulation. Similarly, in [6], PU spectrum occupancies are estimated directly from observations via energy detection while neglecting the underlying probabilistic observation model; on the other hand, our solution centers around a more realistic setup, i.e., a Hidden Markov Model (HMM) formulation in which the true PU occupancy states are hidden behind noisy observations at an SU's spectrum sensor.

Thirdly, the algorithms in [4]–[11], [13]–[15] fail to provide a mechanism to manage the trade-off between secondary network throughput and PU interference; in contrast, we enable this feature through parameter-tuning in our approximate POMDP model. Contrasting our framework against black-box ML/DL models, it is obvious that our approach involving online estimation

of the MDP transition model (time-frequency PU occupancy correlation structure) concurrent with channel sensing and access, circumvents laborious data collection and pre-processing tasks. Unlike a non-adaptive strategy like the Viterbi algorithm in [7] which employs a fixed channel sensing set throughout its period of operation, our solution adapts the sensing action in each time-step, driven by transition model estimates and reward/penalty feedback. Next, highlighting our solution against the model-free RL model described in [15], i.e., an adaptive Deep Q-Network (DQN) formulation – which frames the problem under an unknown Markovian time-frequency correlated PU occupancy structure – our solution achieves superior performance owing to a more accurate estimation of the transition model parameters and more nuanced approximations based on this correlation in PU occupancy behavior – namely, fragmentation (frequency correlation) and Hamming distance state filters (temporal correlation).

Finally, analyzing the state-of-the-art in the distributed cognitive radio networks domain, we find both collaborative as well as opportunistic schemes for channel sensing and access, namely: [6] describes a multi-agent TD-SARSA framework with LFA, while [13] details a collaborative scheme (greedy learning under pre-allocation) and an opportunistic scheme (g-statistics with ACKs). However, [6] fails to detail neighbor discovery and channel access order allocation schemes; the framework in [13] requires *a priori* knowledge of the steady state occupancy probabilities of the channels in the discretized spectrum of interest; additionally, the opportunistic scheme in [13] relies on ACKs as a feedback mechanism from the radio environment to gauge the utility of an access decision, which imbibes unnecessary lag into the model. On the other hand, our framework employs a threshold-based decision heuristic involving the posterior belief probability to evaluate the reward obtained from the executed access action: in addition to displaying superior performance, as illustrated in Sec. IV, this mechanism is easier to implement in real-world settings, as we demonstrate by realizing our solution on the DARPA Spectrum Collaboration Challenge (SC2) emulator [16] and on a custom-built ESP32 WLAN [17] network.

Contributions: In a nutshell, the contributions of this paper are itemized below.

- We develop a POMDP formulation for spectrum sensing and access in a radio environment with PUs exhibiting Markovian correlation in their occupancy behavior across both time and frequency, and an SU trying to adaptively detect and access unused spectral resources – under a noisy observation model with sensing restrictions.
- In pursuit of a solution to this problem, we develop an online parameter estimation algorithm to learn the PUs' occupancy correlation model via an Expectation-Maximization (EM)

scheme for HMMs – namely, the Baum-Welch algorithm; and

- Concurrently, we leverage these learned statistics in a randomized point-based value iteration algorithm known as PERSEUS to devise an approximately optimal spectrum sensing and access policy; additionally, we alleviate its computational complexity by introducing fragmentation and belief update simplification heuristics via Hamming distance state filters. Through computational time complexity bench-marking, we demonstrate the superior scalability of this framework as opposed to comparable works in the state-of-the-art.
- Next, we extend this single-agent formulation to distributed multi-agent deployments – with neighbor discovery (RSSI thresholding) and channel access rank allocation (quorum-based preferential ballot) – and demonstrate enhanced performance over both collaborative and opportunistic distributed multi-agent state-of-the-art; also, we exemplify its implementation feasibility on an ad-hoc WLAN testbed of ESP32 radios [17], [18].
- In order to evaluate the performance of our POMDP policy in centralized multi-agent settings, we retrofit it into our DARPA SC2 BAM! Wireless radio [19] to emulate its operations during the Active Incumbent scenario (TDWR-UNII WLAN) [20], and prove superior performance over different competing strategies revolving around a weighted PSD+CIL approach [21]–[25]. We also perform computational time complexity analyses of our neighbor discovery and channel access rank allocation heuristics in emulations of highly-mobile real-world disaster relief (SC2 Payline [26]) and military deployment scenarios (SC2 Alleys of Austin [27]).
- Furthermore, vis-à-vis single-agent formulations, in addition to proving superior performance over TD-SARSA with LFA, adaptive DQN, correlation-based clustering, and other works in the state-of-the-art – the effects of which are even more striking with our incorporation of optimized rate adaptation [28], [29], we demonstrate that our approximate solution achieves an extremely low normalized sub-optimality gap of 5%; finally, we illustrate our framework's much-needed capability of tuning the trade-off between SU throughput and PU interference.

The rest of this paper is organized as follows: Sec. II details the system model; Sec. III describes our algorithmic solutions; Sec. IV presents numerical evaluations for the single-agent case; Sec. V includes an extension of our solution to a distributed multi-agent setup, which is implemented on a decentralized ad-hoc WLAN testbed of ESP32 radios – followed by a

centralized multi-agent deployment in a DARPA SC2 Active Incumbent scenario emulation; finally Sec. VI provides concluding remarks.

II. System Model

A. Signal Model

We consider a primary network of J PUs, and a secondary network of $\tilde{J}$ SUs exploiting portions of the spectrum left unused by these PUs – as illustrated in Fig. 1. In Sec. II, III, and IV, we focus on the single-agent case ($\tilde{J}$=1); we discuss the multi-agent case ($\tilde{J}$>1) in Sec. V. The spectrum of interest is discretized into K channels of equal bandwidth W. The discretized wide-band signal received at the SU's spectrum sensor in time-slot i at carrier frequency k can be expressed in the frequency domain as

$$Y_k(i) = \sum_{j=1}^{J} H_{j,k}(i) X_{j,k}(i) + V_k(i), \tag{1}$$

where $X_{j,k}(i)$ represents the frequency domain signal of PU $j \in \{1, 2, \ldots, J\}$ in channel $k \in \{1, 2, \ldots, K\}$, with $X_{j,k}(i)=0$ if PU j is not transmitting over channel k in time-slot i; $H_{j,k}(i)$ denotes the frequency domain channel (indexed by k) between the SU and PU j; and $V_k(i) \sim \mathcal{CN}(0, \sigma_V^2)$ constitutes the zero-mean circularly symmetric additive complex Gaussian noise with variance σ_V^2, i.i.d across time and frequency, and independent of the channel H and the PU signal X. Assuming an Orthogonal Frequency Division Multiple Access (OFDMA) strategy among the PUs, and letting $X_k(i) \triangleq X_{j_{k,i},k}(i)$ and $H_k(i) \triangleq H_{j_{k,i},k}(i)$, where subscript $j_{k,i}$ denotes the index of the PU that occupies channel k in time-slot i, we can rewrite (1) as

$$Y_k(i) = H_k(i) X_k(i) + V_k(i), \tag{2}$$

where $X_k(i)=0$ if channel k is idle in time-slot i. We model the frequency domain channel as Rayleigh fading with variance σ_H^2: $H_k(i) \sim \mathcal{CN}(0, \sigma_H^2)$, i.i.d across time and frequency.

B. Occupancy Correlation Structure

The frequency domain signal of the PU occupying channel k in time-slot i is modeled as

$$X_k(i) = \sqrt{P_T} B_k(i) S_k(i), \tag{3}$$

where P_T denotes the transmission power of the occupant PU; $B_k(i)$ represents the binary channel occupancy variable, with $B_k(i)=1$ if channel k is occupied by a PU in time-slot i, and

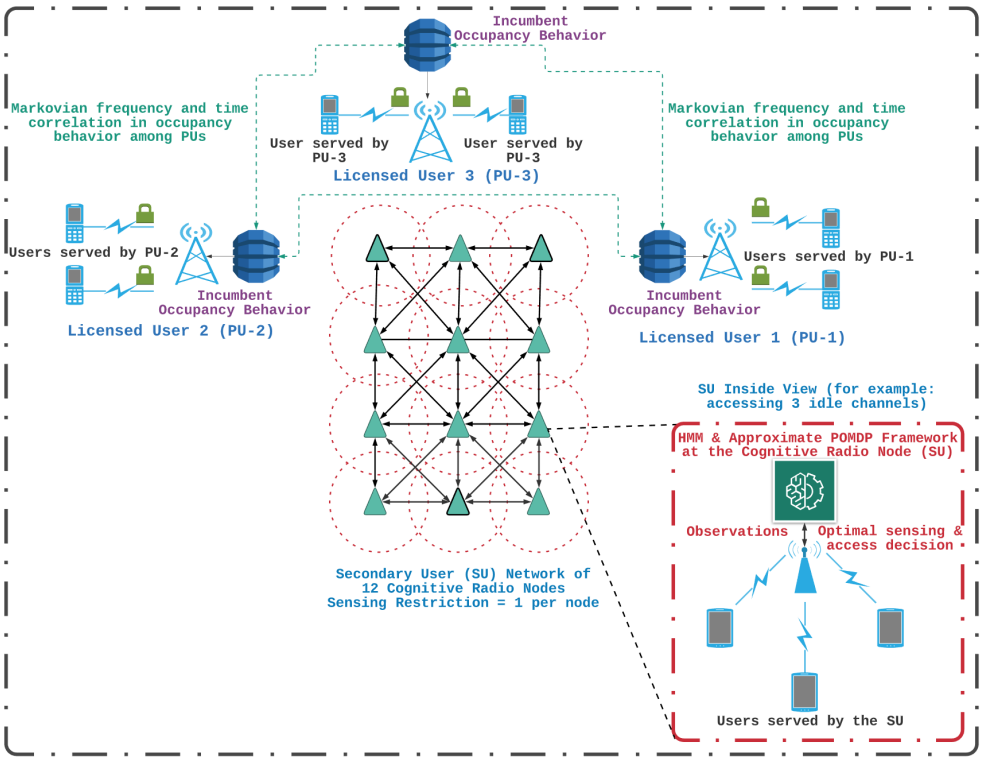

Fig. 1. The radio ecosystem under analysis: An exemplification of the system model detailed in Sec. II-A with J=3 and $\tilde{J}$=12: we first study deployment scenarios with $\tilde{J}$=1 before extending our analysis to multi-agent settings

$B_k(i)$=0 otherwise; $S_k(i)$ is the transmitted symbol, i.i.d across time and frequency, modeled from a certain constellation. Then, $H_k(i)X_k(i)$=$\sqrt{P_T}B_k(i)H_k(i)S_k(i)$. Herein, we approximate $H_k(i)S_k(i)$ as a zero-mean complex Gaussian random variable with variance $\sigma_H^2\mathbb{E}[|S_k|^2]$. We denote the spectrum occupancy state in time-slot i as

$$\vec{B}(i) = [B_1(i), B_2(i), B_3(i), \ldots, B_K(i)]^\intercal \in \{0,1\}^K. \tag{4}$$

We assume that spectrum occupancy is correlated in time and frequency: PUs typically occupy a set of adjacent channels (frequency correlation), repeating similar motifs in behavior over an extended period of time (temporal correlation) [12], [30], [31]. To capture temporal correlation, we model the evolution of $\vec{B}(i)$ over time as a Markov process

$$\mathbb{P}(\vec{B}(i+1)|\vec{B}(j), \forall j \leq i) = \mathbb{P}(\vec{B}(i+1)|\vec{B}(i)). \tag{5}$$

Moreover, to model frequency correlation, we further decompose $\mathbb{P}(\vec{B}(i+1)|\vec{B}(i))$ as

$$\mathbb{P}(\vec{B}(i+1)|\vec{B}(i)) = \mathbb{P}(B_1(i+1)|B_1(i)) \prod_{k=2}^{K} \mathbb{P}(B_k(i+1)|B_{k-1}(i+1), B_k(i)). \tag{6}$$

In other words, the occupancy of frequency band k in time-slot i+1 depends on the occupancy of the adjacent frequency band k−1 in the same time-slot i+1, and that of the same frequency

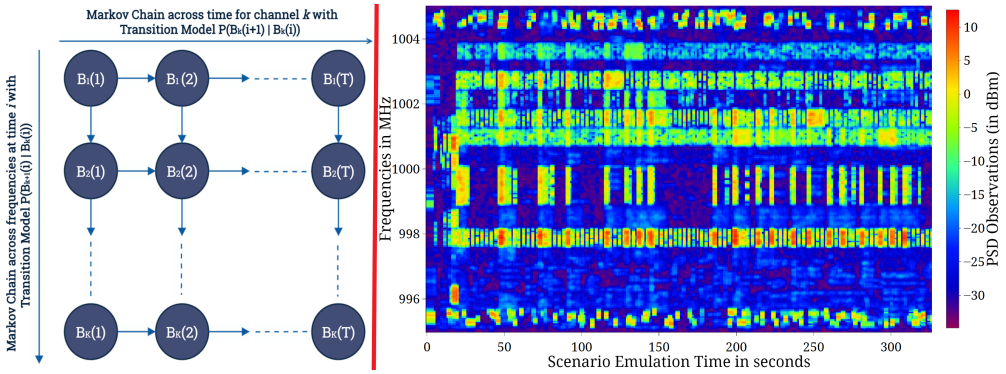

Fig. 2. The visualization of the PU occupancy time-frequency correlation structure as two Markov chains: one across time and the other across frequencies (L) | The combined PSD plot of the occupancy behavior of the PU (TDWR) and the competitors during the DARPA SC2 Active Incumbent scenario emulation (R)

band k in the previous time-slot i, as illustrated in Fig. 2 (L). If the frequency correlation direction is changed, i.e., the occupancy of channel $k{+}1$ influences the occupancy of channel k, $k{\in}\{1,2,\ldots,K{-}1\}$ (bottom-up vs top-down correlation), our model and subsequent analyses still hold: mathematically, the *reversibility* of this Markov chain across frequency – derived from (6) using the Bayes' Rule [32] – can be written as

$$\mathbb{P}(\vec{B}(i{+}1)|\vec{B}(i)){=}\mathbb{P}(B_K(i{+}1)|B_K(i))\prod_{k=1}^{K-1}\mathbb{P}(B_k(i{+}1)|B_{k+1}(i{+}1),B_k(i)). \tag{7}$$

Without loss of generality, sticking with frequency correlation in the forward direction – described by (6) – we parameterize the two-chain Markovian correlation structure underlying PU occupancy behavior with the vector $\vec{\theta}{=}[\vec{p}\ \vec{q}]^\intercal$, where

$$\begin{aligned}\vec{p} &= [p_{uv} = \mathbb{P}(B_k(i+1)=1|B_{k-1}(i+1)=u, B_k(i)=v) : u,v\in\{0,1\}, 1{<}k{\leq}K, 1{\leq}i{\leq}T]^\intercal;\\ \vec{q} &= [q_w = \mathbb{P}(B_1(i+1)=1|B_1(i)=w) : w\in\{0,1\}, 1{\leq}i{\leq}T]^\intercal.\end{aligned} \tag{8}$$

To experimentally validate the aforementioned parameterized time-frequency correlation model, we perform Bayesian Information Criterion (BIC) evaluation, i.e.,

$$BIC = \Gamma\ln\nu - 2\ln\mathbb{P}(\mathbf{B}|\hat{\vec{\theta}}^*), \tag{9}$$

on a dataset constituting PSD measurements of a Terminal Doppler Weather Radar (TDWR) PU and several competitor secondary networks [24], [25] (including the Purdue BAM! Wireless Network [19]) constituting Unlicensed National Information Infrastructure (UNII: 5GHz WLAN) Wi-Fi nodes, in the DARPA SC2 Active Incumbent scenario emulation [16], [20] – depicted in Fig. 2 (R). In (9), ν is the sample size, Γ is the number of model parameters, $\mathbf{B}$ is the time-frequency binary occupancy matrix from the dataset, and $\hat{\vec{\theta}}^*$ consists of the correlation model

parameters estimated from this dataset of PSD observations, using the Baum-Welch algorithm detailed in Sec. III-A. We employ a $70{-}30$ training-test split to evaluate this metric, i.e., the occupancy data collected during the first 70% of the 330 seconds of scenario emulation is used to estimate the model parameters, while the remaining 30% is dedicated to determining the BIC metric for these estimates. Our proposed time-frequency correlation model yields a BIC of 71.872, the best compared to other state-of-the-art models: time-frequency independence models (98.840) [5], [8]–[11], only temporal correlation models (95.231) [6], and only frequency correlation models (76.879). This evaluation reveals that exploiting frequency correlation is more important than time correlation – and not surprisingly, exploiting *both* time and frequency correlation provides a better fit to the dataset. This assessment validates our hypothesis that PUs in real-world radio deployments exhibit prominent patterns in their occupancy behavior across both time and frequency. Additionally, the time-frequency Markovian correlation model considering a bottom-up correlation across frequency, i.e., a reversed Markov chain across frequency – described by (7) – can be parameterized by $\vec{\theta}_r{=}[\vec{p}_r\ \vec{q}_r]^\intercal$, where

$$\begin{aligned}\vec{p}_r &= [p_{uv} = \mathbb{P}(B_k(i+1)=1 | B_{k+1}(i+1)=u, B_k(i)=v) : u,v \in \{0,1\}, 1{\leq}k{<}K, 1{\leq}i{\leq}T]^\intercal;\\ \vec{q}_r &= [q_w = \mathbb{P}(B_K(i+1)=1 | B_K(i)=w) : w \in \{0,1\}, 1{\leq}i{\leq}T]^\intercal.\end{aligned} \tag{10}$$

In order to empirically validate our postulation that a reversed Markovian direction across frequency captures the same amount of correlation in PU occupancy behavior, we fit the model detailed in (7) to the aggregated PSD measurements from the same DARPA SC2 Active Incumbent scenario (same $70{-}30$ training-test split), estimate the associated parametric model $\hat{\vec{\theta}}_r^*$ using the Baum-Welch algorithm, and obtain a BIC metric of 74.207 – thereby corroborating that our model and subsequent analyses hold under a bottom-up correlation across frequency. Note that we do not consider frequency correlation in *both* forward and backward directions because our framework then loses its Markovian properties, which preempts us from employing an HMM-POMDP formulation; also, in practice, Markovian correlation in a single direction across frequency – coupled with a temporal Markov chain – sufficiently approximates patterns in PU occupancy behavior.

C. Channel Sensing Model

Equipped with a spectrum sensor, the SU detects white-spaces and accesses them to deliver its network flows. Due to constraints on energy-efficiency and sensing/data aggregation times

[4], the SU can sense a maximum of κ spectrum bands in a time-slot, with $1 \leq \kappa \leq K$. Let $\mathcal{K}_i \subseteq \{1, 2, \ldots, K\}$ be the set of channels sensed by the SU at time i, with $|\mathcal{K}_i| \leq \kappa$. This selection is dictated by a sensing policy, as described in Sec. III-B. After sensing the channels listed in $\mathcal{K}_i$, the obtained observation vector is $\vec{Y}(i) = [Y_k(i)]_{k \in \mathcal{K}_i}]$, with $Y_k(i)$ given in (2). Owing to (2), the i.i.d. assumptions of the noise $V_k(i)$, the transmitted symbols $S_k(i)$, and the frequency domain channels $H_k(i)$, the probability density function (pdf) of $\vec{Y}(i)$ conditioned on the spectrum occupancy vector $\vec{B}(i)$ and the sensing set $\mathcal{K}_i$ is given by

$$f(\vec{Y}(i)|\vec{B}(i), \mathcal{K}_i) = \prod_{k=1}^{K} f(Y_k(i)|B_k(i)), \text{ where } Y_k(i)|B_k(i) \sim \mathcal{CN}(0, \sigma_H^2 P_T B_k(i) + \sigma_V^2). \quad (11)$$

D. POMDP Formulation

POMDPs model the repeated, sequential interactions of an agent tasked with maximizing its reward, with a stochastic environment, in which the agent has only access to noisy observations of the state. Our POMDP formulation, depicted in Fig. 3 and represented by the 5-tuple $(\mathcal{B}, \mathcal{A}, \mathcal{Y}, \mathbf{A}, \mathbf{M})$, features the state space $\mathcal{B} \equiv \{0, 1\}^K$ of the underlying MDP, given by all possible realizations of the occupancy vector $\vec{B}$; the action space $\mathcal{A}$ of the SU, described by all possible combinations in which $1 \leq \kappa \leq K$ channels are chosen to be sensed in a time-slot (discussed in Sec. II-C); the observation space $\mathcal{Y}$, discussed in Sec. II-A; the transition model $\mathbf{A}$ of the underlying MDP, discussed in Sec. II-B; and the observation model $\mathbf{M}$, described by (11). The POMDP process flow – relevant to our discussions in this section – is illustrated in Fig. 3.

Prior to gathering the occupancy information in time-slot i, based on the measurements obtained by the SU's spectrum sensor up to, but not including, time-slot i, the POMDP state is given by the prior belief β_i, representing the probability distribution of the underlying MDP state $\vec{B}(i)$ given the history. Given β_i, the SU chooses a sensing action according to a sensing policy $\mathcal{K}_i = \pi(\beta_i) \in \mathcal{A}$, senses the frequency bands corresponding to the channel indices in the set $\mathcal{K}_i$ (Sec. II-C), observes $[Y_k(i)]_{k \in \mathcal{K}_i} \in \mathcal{Y}$, and computes the posterior belief of $\vec{B}(i)$ as

$$\begin{aligned} \hat{\beta}_i(\vec{B}') &= \mathbb{P}(\vec{B}(i) = \vec{B}' | \beta_i, \mathcal{K}_i, [Y_k(i)]_{k \in \mathcal{K}_i}) \\ &= \frac{\mathbb{P}([Y_k(i)]_{k \in \mathcal{K}_i} | \vec{B}', \mathcal{K}_i) \beta(\vec{B}')}{\sum_{\vec{B}'' \in \{0,1\}^K} \mathbb{P}([Y_k(i)]_{k \in \mathcal{K}_i} | \vec{B}'', \mathcal{K}_i) \beta_i(\vec{B}'')}. \end{aligned} \quad (12)$$

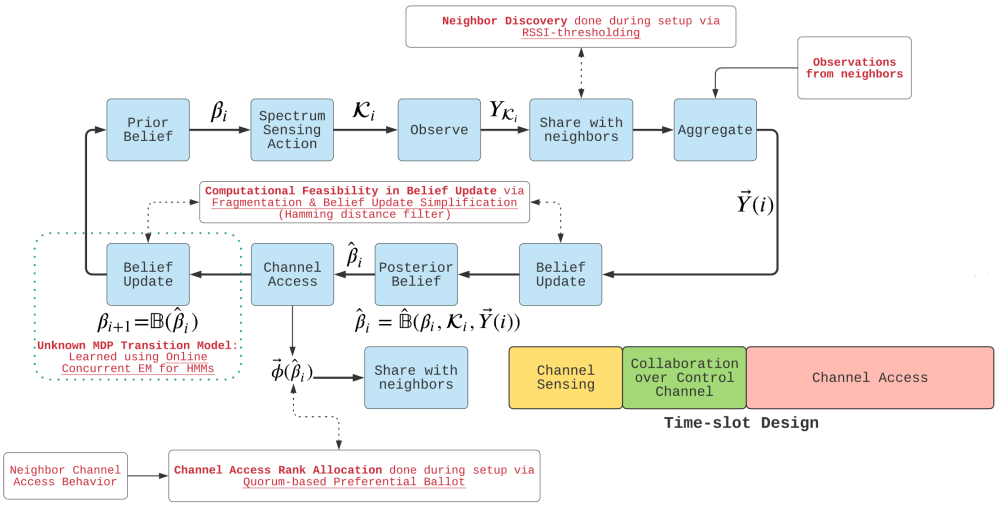

Fig. 3. The POMDP process flow as discussed in Sec. II-D, with neighbor discovery, channel access rank allocation, and time-slot design being relevant design points discussed in our multi-agent deployment analysis (Sec. V)

Given $\hat{\beta}_i$, the SU then performs channel access decisions $\vec{\phi}(i) \in \{0,1\}^K$, where $\vec{\phi}_k(i) = 1$ if the SU accesses the kth spectrum band, and $\vec{\phi}_k(i) = 0$ otherwise. To determine $\vec{\phi}(i)$, we define the reward metric (considering the true occupancy state vector) as

$$R(\vec{\phi}(i), \vec{B}(i)) = \sum_{k=1}^{K} (1 - B_k(i))\phi_k(i) - \lambda \vec{B}_k(i)\phi_k(i), \tag{13}$$

and its expectation based on the current posterior belief as

$$R(\vec{\phi}(i), \hat{\beta}_i) = \mathbb{E}[R(\vec{\phi}(i), \vec{B}(i))|\vec{\phi}(i), \hat{\beta}_i] = \sum_{k=1}^{K} (1 - \hat{\beta}_{i,k})\phi_k(i) - \lambda \hat{\beta}_{i,k}\phi_k(i). \tag{14}$$

Note that, if the SU uses the kth spectrum band ($\phi_k(i) = 1$), it accrues a reward if the spectrum band is truly idle (with posterior probability $1 - \hat{\beta}_{i,k}$), and a penalty λ if it is occupied (with posterior probability $\hat{\beta}_{i,k}$); it accrues no reward for not using a channel. Hence, this reward metric captures both the number of truly idle channels (correctly estimated idle) accessed by the SU, accounting for the throughput maximization aspect of our objective, as well as the number of truly occupied channels (incorrectly estimated idle) accessed by it, accounting for the PU interference minimization aspect of our objective, where λ regulates such trade-off. The channel access decision is $\vec{\phi}^*(i) = \arg\max R(\vec{\phi}(i), \hat{\beta}_i)$, given in closed form as

$$\phi_k^*(i) = \begin{cases} 1, & \hat{\beta}_i < \frac{1}{1+\lambda} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

yielding the reward

$$R^*(\hat{\beta}_i) = \max_{\vec{\phi} \in \{0,1\}^K} R(\vec{\phi}(i), \hat{\beta}_i) = \sum_{k=1}^{K} \max\{1 - (1+\lambda)\hat{\beta}_{i,k}, 0\}.$$

In other words, if the agent is confident that the channel is idle ($\hat{\beta}_i < \frac{1}{1+\lambda}$), then the SU accesses it; otherwise, it remains idle.

Ensuing the determination of the reward for its access decision from the radio environment, the SU computes the prior belief for the next time-slot $i+1$ as

$$\beta_{i+1}(\vec{B}'') = \sum_{\vec{B}'} \mathbb{P}(\vec{B}(i+1) = \vec{B}''|\vec{B}(i) = \vec{B}')\hat{\beta}_i(\vec{B}'), \tag{16}$$

and the process is repeated over time. Let

$$\hat{\beta}_i = \hat{\mathbb{B}}(\beta_i, \mathcal{K}_i, \vec{Y}(i)), \quad \beta_{i+1} = \mathbb{B}(\hat{\beta}_i) \tag{17}$$

denote the functions that map the prior belief β_i to the posterior belief $\hat{\beta}_i$ in time-slot i, and the posterior belief $\hat{\beta}_i$ to the next prior belief β_{i+1} in time-slot $i+1$. The objective of the SU is to devise a spectrum sensing policy (based on which the access decisions are made in the corresponding time-slots) to maximize its infinite-horizon discounted reward – albeit with approximations and heuristics to render the approach computationally scalable, i.e.,

$$\pi^* = \arg\max_{\pi} V^{\pi}(\beta), \tag{18}$$

where

$$V^{\pi}(\beta) = \mathbb{E}_{\pi}\left[\sum_{i=1}^{\infty} \gamma^i R^*(\hat{\beta}_i) \Big| \beta_0 = \beta\right], \tag{19}$$

$0<\gamma<1$ is the discount factor, $\beta_0=\beta$ is the initial belief, and $\hat{\beta}_i$ is the posterior belief induced by the policy $\mathcal{K}_i=\pi(\beta_i)$ and the observation vector $[Y_k(i)]_{k\in\mathcal{K}_i}$ via $\hat{\beta}_i = \hat{\mathbb{B}}(\beta_i, \mathcal{K}_i = \pi(\beta_i), [Y_k(i)]_{k\in\mathcal{K}_i})$. Theoretically, the optimal value function $V^*(\beta)$ can be shown to be solution of the Bellman's optimality equation $V^*=\mathcal{H}(V^*)$ [33], where $\mathcal{H}$ is the Bellman's operator, defined as $V_{t+1}=\mathcal{H}(V_t)$ with

$$V_{t+1}(\beta) = \max_{\mathcal{K}\in\mathcal{A}} \sum_{\vec{B}\in\mathcal{B}} \beta(\vec{B}) \mathbb{E}_{[Y_k]_{k\in\mathcal{K}}|\vec{B},\mathcal{K}} \left[R^*(\hat{\mathbb{B}}(\beta, \mathcal{K}, [Y_k]_{k\in\mathcal{K}})) + \gamma V_t(\mathbb{B}(\hat{\mathbb{B}}(\beta, \mathcal{K}, [Y_k]_{k\in\mathcal{K}})))\right], \ \forall \beta. \tag{20}$$

V^* can be determined via the value iteration algorithm $V_{t+1} = \mathcal{H}(V_t)$, which converges to V^* as $t \to \infty$ [33]. However, this direct approach results in complications associated with the lack of prior knowledge about the PU occupancy time-frequency correlation structure that defines the transition model of the underlying MDP, and the computational infeasibility of the approach: as the number of channels in the discretized spectrum of interest increases, the number of states of the underlying MDP scales exponentially, resulting in a high-dimensional belief space. To address these two challenges, we propose the following solutions:

- We incorporate an HMM EM estimator, i.e., the Baum-Welch algorithm, to learn the time-frequency occupancy correlation structure while concurrently solving for the sensing and access policy. This is developed in Sec. III-A.
- We embed a low-complexity approximate value iteration algorithm known as PERSEUS [34], with fragmentation (into independent subsets of highly-correlated channels) and belief update simplification heuristics (Hamming distance state filters), developed in Sec. III-B.

III. Proposed Solution: The Algorithms

Practical MAC layer implementations of cognitive radios involve solving for the sensing and access policy, without having any prior information about the time-frequency correlation structure underlying the occupancy behavior of the PUs in the network. [23], [35]. As discussed earlier, this correlation structure may be leveraged to improve white-space detection, hence utilization. In this section, we propose a parameter estimator algorithm that learns this correlation structure over time. In Sec. III-B, we then use this knowledge in an randomized point-based value iteration framework based on PERSEUS [34] to determine an approximately optimal sensing and access policy. Crucially, the parameter estimation and PERSEUS algorithms are executed concurrently, which is especially vital in non-stationary settings.

A. Occupancy Correlation Structure Estimation

Let τ refer to the learning period of the parameter estimation algorithm: this may be equal to the entire duration of the SU's interaction with the radio environment while solving for the sensing and access policy, implying concurrent model learning, or it can be equal to an initial learning period that has been set aside exclusively for the SU to estimate the underlying MDP's transition model, after which the PERSEUS algorithm is initiated, employing these final estimated (converged) transition probabilities. Defining $\mathbf{B}=[\vec{B}(i)]_{i=1}^{\tau}$ as the unknown sequence of states and $\mathbf{Y}=[\vec{Y}(i)]_{i=1}^{\tau}$ as the sequence of observations made at the SU's spectrum sensor from $i=1$ to $i=\tau$, we formulate the Maximum Likelihood Estimation (MLE) problem to estimate the vector $\vec{\theta}$ that parameterizes the PU occupancy time-frequency correlation structure (detailed in Sec. II-B) as

$$\vec{\theta}^{*} = \arg\max_{\vec{\theta}} \log \left(\sum_{\mathbf{B}} \mathbb{P}(\mathbf{B}, \mathbf{Y}|\vec{\theta}) \right). \tag{21}$$

Solving this MLE formulation using the Baum-Welch algorithm, an Expectation-Maximization algorithm for HMMs [36], the E-step constitutes

$$Q(\vec{\theta}|\vec{\theta}^{(t)}) = \mathbb{E}_{\mathbf{B}|\mathbf{Y},\vec{\theta}^{(t)}}\left[\log\left(\mathbb{P}(\mathbf{B},\mathbf{Y}|\vec{\theta}^{(t)})\right)\right], \tag{22}$$

which can be computed using the Forward-Backward algorithm [37]; and the M-step constitutes

$$\vec{\theta}^{(t+1)} = \arg\max_{\vec{\theta}} Q(\vec{\theta}|\vec{\theta}^{(t)}), \tag{23}$$

which involves the re-estimation of $\vec{\theta}$ by employing the statistics $Q(\vec{\theta}|\vec{\theta}^{(t)})$ obtained from the Forward-Backward algorithm [37].

B. The PERSEUS Algorithm

In our proposed solution, we solve for the spectrum sensing (and access, based on reward maximization detailed in Sec. II-D) policy, in parallel with the parameter estimation algorithm, employing its published iterative transition model estimates, until both the EM algorithm and the POMDP policy solver algorithms converge. As alluded to in Sec. II-D, in order to solve the computational infeasibility caused by the exponential increase in the number of states of the underlying MDP, induced by an increase in the number of frequency bands in the discretized spectrum of interest, we employ approximate POMDP value iteration methods to ensure that the formulations and the algorithms scale well to a large number of relevant channels in the radio environment in which the SU operates. We choose the PERSEUS algorithm [34] to unravel our POMDP formulation and devise an approximately optimal sensing and access policy, primarily motivated by the following: unlike the Exhaustive Enumeration algorithm and the Witness algorithm in [33], the PERSEUS algorithm does not involve performing the backup operation for every point in the belief space; and unlike the Point-Based Value Iteration (PBVI) algorithm in [38], it does not require computing belief distances, and does not involve performing backups on all the reachable belief points; instead, PERSEUS involves *backing-up* only on a subset of this set of reachable beliefs, while ensuring that the computed solution is effective for all the points in the reachable belief set, yielding lower computational complexity. The PERSEUS algorithm, although is an approximate POMDP method which eliminates the computational overhead associated with the exhaustive belief space and reachable space optimization techniques [33], [38] by approximating the optimization of a randomly chosen belief point to the entire set of unimproved, reachable belief points, still possesses computational intractability challenges

because it involves iterations over all possible combinations of the occupancy state vector, i.e., $\vec{B}\in\{0,1\}^K$: the computational cost scales doubly-exponentially with the number of channels K in the discretized spectrum of interest. In order to solve this computational tractability problem, we introduce two simplifying heuristics into the PERSEUS algorithm.

Firstly, we avoid iterating over all possible occupancy states by considering only those state transitions that involve a Hamming distance of $\delta\in\{1,2,\ldots,K\}$ between two consecutive state vectors, $\vec{B}$ and $\vec{B}'$. This is practical because the temporal dynamics governing the spectrum occupancies, dictated by the behavior of the PUs in the network, are typically slower than the processing dynamics of the POMDP agent: mathematically, for a state $\vec{B}$, the Hamming distance filtered state space for probable consecutive states $\vec{B}'$ is given by $\mathcal{B}_\delta(\vec{B})\equiv\{\vec{B}'\in\mathcal{B}:\zeta(\vec{B},\vec{B}')\leq\delta\}$, where ζ denotes the Hamming distance between the two vectors.

Secondly, we fragment the discretized spectrum into smaller, independent sets of correlated channels (for example, an 18 channel radio environment with 3 PUs and 1 SU with a sensing restriction of 6 channels per time-slot, is fragmented into 3 independent fragments, each comprising 6 channels correlated by the occupancy behavior of the corresponding PU, and the SU restricted to sensing 2 channels per fragment per time-slot); run PERSEUS on these fragments concurrently by employing multi-threading capabilities in software frameworks; and finally, combine the results from each of these fragmented, parallel runs to get a full picture about the performance of our POMDP agent. This is practical because in a radio environment with multiple PUs, each PU is typically restricted to a portion (a set of adjacent frequency bands) of the spectrum, either by design or by bureaucracy: mathematically, we represent the POMDP model for a fragment indexed by g of size $\Delta_g=\Delta\in\{1,2,\ldots,K\},\ \sum_g\Delta_g=K$ as $(\mathcal{B}_\Delta,\mathcal{A}_\Delta,\mathcal{Y}_\Delta,\mathbf{A}_\Delta,\mathbf{M})$ with a sensing restriction $\kappa_{\Delta_g}=\kappa_\Delta$, where $\mathcal{B}_\Delta\equiv\{0,1\}^\Delta$ is its state space dependent on its transition model $\mathbf{A}_\Delta$, $\mathcal{Y}_\Delta$ is its observation space dependent on a common observation model $\mathbf{M}$, and its action space $\mathcal{A}_\Delta$ corresponds to the set of all possible combinations in which $1\leq\kappa_\Delta\leq\kappa$ channels are chosen to be sensed in a time-slot, such that $\sum_g\kappa_{\Delta_g}=\kappa$.

Employing this fragmented POMDP model – along with additional utilities for a Hamming distance state filter, prior and posterior belief updates, and value function evaluation – PERSEUS involves an initial phase of exploration, wherein the set of *reachable-beliefs*, denoted by $\tilde{\mathcal{B}}$, is determined by allowing the SU to randomly interact with the radio environment (Step 1 in Alg. 1). As referenced earlier, one simplifying (or approximating) feature of PERSEUS is to improve the value of all the belief points in the set $\tilde{\mathcal{B}}$, by computing the value of only a subset of these

Algorithm 1 Fragmented PERSEUS with Belief Update Simplification

Fragmented POMDP Model: $(\mathcal{B}_\Delta, \mathcal{A}_\Delta, \mathcal{Y}_\Delta, \mathbf{A}_\Delta, \mathrm{M})$ ▷ $\mathbf{A}_\Delta$ parameterized by $\hat{\vec{\theta}}$

Utilities: Hamming distance state filter: $\mathcal{B}_\delta(\vec{B}) \equiv \{\vec{B}' \in \mathcal{B} : \zeta(\vec{B}, \vec{B}') \leq \delta\}$;

Posterior belief: $\hat{\mathbb{B}}(\beta_i, \mathcal{K}_i, \vec{Y}(i)) = \hat{\beta}_i$, where $\hat{\beta}_i(\vec{B}') = \dfrac{\mathbb{P}([Y_k(i)]_{k\in\mathcal{K}_i}|\vec{B}', \mathcal{K}_i)\beta(\vec{B}')}{\sum_{\vec{B}''\in\{0,1\}^K} \mathbb{P}([Y_k(i)]_{k\in\mathcal{K}_i}|\vec{B}'', \mathcal{K}_i)\beta_i(\vec{B}'')}$;

Next prior belief: $\mathbb{B}(\hat{\beta}_i) = \beta_{i+1}$, where $\beta_{i+1}(\vec{B}'') = \sum_{\vec{B}'\in\mathcal{B}_\delta(\vec{B}'')} \mathbb{P}(\vec{B}(i+1) = \vec{B}''|\vec{B}(i) = \vec{B}', \hat{\vec{\theta}})\hat{\beta}_i(\vec{B}')$;

Value function: $V_t(\beta) \approx \beta \cdot \vec{\alpha}_t^{u^*}$, where $u^* = \underset{u\in\{1,2,\ldots,|\tilde{\mathcal{B}}|\}}{\arg\max}\ \beta \cdot \vec{\alpha}_t^u$, $\beta \cdot \vec{\alpha} = \sum_{\vec{B}} \beta(\vec{B})\vec{\alpha}(\vec{B}), \forall \beta \in \tilde{B}$.

Output: The approximately optimal sensing policy π^*, i.e., $\pi^* = \arg\max_\pi V^\pi(\beta)$.

Determine the set of *reachable beliefs* $\tilde{\mathcal{B}}$. ▷ Random exploration

$V_0(\beta) = \frac{-K\lambda}{1-\gamma}, \forall \beta \in \tilde{\mathcal{B}}$. ▷ Initialization

while $|V_{t+1}(\beta) - V_t(\beta)| < \epsilon, \forall \beta \in \tilde{\mathcal{B}},\ \epsilon > 0,$ Iteration index$= t$ **do**

  $\tilde{\mathcal{U}} \longleftarrow \tilde{\mathcal{B}}$ ▷ Set copy for local updates

  while $\tilde{\mathcal{U}} \neq \{\}$ **do** ▷ Improve all points in $\tilde{\mathcal{U}}$

    $\beta_u =$ random.choice$(\tilde{\mathcal{U}})$; $\vec{\alpha}_{t+1}^u = \xi_{\mathcal{K}_{t+1}^u}^u$; $\mathcal{K}_{t+1}^u = \arg\max_{\mathcal{K}\in\mathcal{A}} \beta_u \cdot \xi_{\mathcal{K}}^u$, where ▷ Backup

    $\xi_{\mathcal{K}}^u(\vec{B}) = \mathbb{E}_{\vec{Y}|\vec{B},\mathcal{K}}\Big[R(\hat{\mathbb{B}}(\beta_u, \mathcal{K}, \vec{Y})) + \gamma \sum_{\vec{B}'} \mathbb{P}(\vec{B}(i+1) = \vec{B}'|\vec{B}(i) = \vec{B})\xi_{\mathcal{K},\vec{Y}}^u(\vec{B}')\Big]$, and

    $\xi_{\mathcal{K},\vec{Y}}^u = \arg\max_{\alpha_t^{u'}, u'\in\{1,2,\ldots,|\tilde{\mathcal{B}}|\}} \mathbb{B}(\hat{\mathbb{B}}(\beta_u, \mathcal{K}, \vec{Y})) \cdot \alpha_t^{u'}$. ▷ Future hyperplane

    for $\beta' \in \tilde{\mathcal{U}}$ **do**

      $V_{t+1}(\beta') := V_t(\beta')$; $\vec{\alpha}_{t+1}(\beta') := \vec{\alpha}_t(\beta')$; $\mathcal{K}_{t+1}(\beta') := \mathcal{K}_t(\beta')$. ▷ Persist

      if $\beta' \cdot \vec{\alpha}_{t+1}^u \geq V_t(\beta')$ **then** ▷ Improvement check and subsequent updates

        $V_{t+1}(\beta') = \beta' \cdot \vec{\alpha}_{t+1}^u$; $\vec{\alpha}_{t+1}(\beta') := \vec{\alpha}_{t+1}^u$; $\mathcal{K}_{t+1}(\beta') := \mathcal{K}(\vec{\alpha}_{t+1}^u) \in \mathcal{A}_\Delta$; $\tilde{\mathcal{U}} \longleftarrow \tilde{\mathcal{U}} - \beta'$.

      end if

    end for

  end while

end while

belief points, which are chosen iteratively at random. For finite-horizon POMDP formulations, the optimal value function V^* described by (20), can be approximated by a Piece-Wise Linear Convex (PWLC) function [34] parameterized by a set of hyperplanes $\{\vec{\alpha}_t^u\}, u \in \{1, 2, \ldots, |\tilde{\mathcal{B}}|\}$, wherein each hyperplane represents a region of the belief space for which the action corresponding to

this hyperplane, denoted by $\mathcal{K}_t^u$, is the maximizer. Ergo, the value function of belief β in a given iteration t is approximated as $V_t(\beta) \approx \beta \cdot \max_{u \in \{1,2,\ldots,|\tilde{\mathcal{B}}|\}} \vec{\alpha}_t^u$, where $\beta \cdot \vec{\alpha} = \sum_{\vec{B}} \beta(\vec{B}) \vec{\alpha}(\vec{B})$ denotes inner product. The approximately optimal spectrum sensing action is the one associated with the maximizing hyperplane α^*, and denoted as $\mathcal{K}_t^{u^*}$.

We initialize the value functions corresponding to the reachable beliefs in $\tilde{\mathcal{B}}$ to the minimum of all possible cumulative discounted rewards achievable by the POMDP agent in the given formulation, i.e., $V_0(\beta) = \frac{-K\lambda}{1-\gamma}, \forall \beta \in \tilde{\mathcal{B}}$ [39] (Step 2 in Alg. 1): this initial value is guaranteed to be below V^* [34]. Defining a new set $\tilde{\mathcal{U}} \equiv \tilde{\mathcal{B}}$ for local updates (Step 4 in Alg. 1), we pick a belief β_u from it at random, and perform the backup operation on this chosen belief point, which as discussed earlier, involves associating a new hyperplane and its corresponding spectrum sensing action with this belief β_u (Steps 6, 7, and 8 in Alg. 1): in iteration $t+1$, defining $\mathcal{K}_{t+1}^u$ as the action associated with hyperplane $\vec{\alpha}_{t+1}^u$, the backup procedure is described mathematically in Step 6 of Alg. 1, where $\xi_{\mathcal{K}}^u$ is the hyperplane corresponding to the one-step look-ahead under the considered action $\mathcal{K} \in \mathcal{A}$ for the chosen belief β_u (Step 7 in Alg. 1). Evaluating $\xi_{\mathcal{K}}^u$ involves another operation (Step 8 in Alg. 1) wherein we use the previous set of hyperplanes $(\alpha_t^{u'}, u' \in \{1, 2, \ldots, |\tilde{\mathcal{B}}|\})$ to compute the hyperplane $(\xi_{\mathcal{K},\vec{Y}}^u)$ for the new belief $\mathbb{B}(\hat{\mathbb{B}}(\beta_u, \mathcal{K}, \vec{Y}))$ which is obtained from the current chosen belief β_u by executing the considered action $\mathcal{K} \in \mathcal{A}$ and observing $\vec{Y}$.

After determining the hyperplane α_{t+1}^u associated with this chosen belief point β_u using the backup procedure detailed above, we now know that $V_{t+1}(\beta_u) = \beta_u \cdot \vec{\alpha}_{t+1}^u$ is its approximate value function. The most crucial aspect of PERSEUS is that it uses this new hyperplane to improve the value function of all the remaining belief points in the set of unimproved beliefs $\tilde{\mathcal{U}}$. For a belief point $\beta' \in \tilde{\mathcal{U}}$, it first computes the approximate value function under the new hyperplane $\vec{\alpha}_{t+1}^u$ (Step 11 in Alg. 1). If this value function improves the previously recorded value $V_t(\beta')$, then the new hyperplane generates an improved approximate value function ($V_{t+1}(\beta')$) and a new associated sensing action ($\mathcal{K}_{t+1}(\beta')$), so that β' is removed from the set of unimproved beliefs (Step 12 in Alg. 1). On the other hand, if this hyperplane $\vec{\alpha}_{t+1}^u$ does not improve the approximate value function of β', i.e., $\beta' \cdot \vec{\alpha}_{t+1}^u < V_t(\beta')$: the old hyperplane ($\vec{\alpha}_t(\beta')$) and its associated sensing action ($\mathcal{K}_t(\beta') = \mathcal{K}(\vec{\alpha}_t(\beta')) \in \mathcal{A}_\Delta$) persist for β' (Step 10 in Alg. 1), and we continue to check for improvements with respect to the other belief points in $\tilde{\mathcal{U}}$ (*for* loop in Step 9 of Alg. 1), and remove all those belief points $\beta' \in \tilde{\mathcal{U}}$ for which $\beta' \cdot \vec{\alpha}_{t+1}^u \geq V_t(\beta')$.

In general, if a hyperplane determined from the backup procedure improves a belief point in the set of unimproved belief points $\tilde{\mathcal{U}}$, this news hyperplane (and its associated sensing action)

becomes the relevant hyperplane (and the relevant sensing action) for this belief point, and the belief point will be removed from the set of unimproved belief points $\tilde{\mathcal{U}}$. These sequence of operations (random choice from $\tilde{\mathcal{U}} \longrightarrow$ backup $\longrightarrow$ check for improvement and removal) are performed iteratively until the set $\tilde{\mathcal{U}}$ is empty (*while* loop in Step 5 of Alg. 1): this constitutes a single PERSEUS iteration. These PERSEUS iterations are executed until the specified value iteration termination condition is satisfied, i.e., $|V_{t+1}(\beta) - V_t(\beta)| < \epsilon, \forall \beta \in \tilde{\mathcal{B}}$, where $\epsilon > 0$ (a very small value) is the value iteration difference threshold (*while* loop in Step 3 of Alg. 1).

IV. Numerical Evaluations

Sticking with the single-agent deployment setting, our simulations evaluate the operational capabilities of the proposed POMDP framework and compare it against the state-of-the-art. The simulated radio environment constitutes $J=3$ PUs, i.e., PUs, accessing a 2.88 MHz spectrum, discretized into $K=18$ channels, each having a bandwidth of $W=160$ kHz, and an SU ($\tilde{J}=1$) trying to intelligently access spectrum holes to deliver its network flows while limiting PU interference, as illustrated in Fig. 1. The 3 PUs access these 18 channels according to a time-frequency Markovian correlation structure parameterized by $\vec{\theta} = [\vec{p}; \vec{q}]$, where

$$\vec{p} = \begin{bmatrix} p_{00} = 0.1 & p_{01} = 0.3 & p_{10} = 0.3 & p_{11} = 0.7 \end{bmatrix}, \text{ and}$$

$$\vec{q} = \begin{bmatrix} q_0 = 0.3 & q_1 = 0.8 \end{bmatrix}.$$

We denote the sensing constraint as $\kappa=6$. Regarding the expected Signal to Interference Noise Ratios (SINR) at the PUs and the SU, subject to fading, and conditioned on the PU and SU access decisions, we model our simulation framework based off the following numbers:

$\text{SINR}_{\text{SU}}(k,i)=0$,	if the SU does not access channel k in time-slot i,
$\text{SINR}_{\text{SU}}(k,i)=11$ dB,	if the SU accesses a truly idle channel k in time-slot i,
$\text{SINR}_{\text{SU}}(k,i)=-6$ dB,	if SU accesses a PU-occupied channel k in slot i,
$\text{SINR}_{\text{PU}_j}(k,i)=0$,	if the PU j does not access channel k in time-slot i,
$\text{SINR}_{\text{PU}_j}(k,i)=17$ dB,	if PU j occupies channel k in slot i without SU interference,
$\text{SINR}_{\text{PU}_j}(k,i)=6$ dB,	if PU j occupies channel k in slot i with SU interference.

To evaluate the performance of the proposed scheme, we define the average throughput attained by the SU over T time-slots as

$$C^{\text{SU}} = \frac{1}{T}\sum_{i=1}^{T}\sum_{k=1}^{K} R_{\text{SU}}\mathcal{I}\left\{\text{SINR}_{\text{SU}}(k,i) \geq 2^{\frac{R_{\text{SU}}}{W}} - 1\right\}, \tag{24}$$

where R_{SU}=0.6 Mbps is the transmission rate of the SU on each channel, and $\mathcal{I}$ is an indicator variable; and the throughput attained by the PUs in the network over the same T time-slots, normalized over time and the number of transmissions is given by

$$C^{\text{PUs}} = \frac{\sum_{i=1}^{T}\sum_{k=1}^{K} R_{\text{PU}}B_k(i)\mathcal{I}\left\{\text{SINR}_{\text{PU}}(k,i) \geq 2^{\frac{R_{\text{PU}}}{W}} - 1\right\}}{\sum_{i=1}^{T}\sum_{k=1}^{K} B_k(i)}, \tag{25}$$

where R_{PU}=0.9 Mbps is the transmission rate of the PUs on each channel, and with $j \in \{1, 2, \ldots, J\}$ being the index of the PU occupying channel k in time-slot i, we have $\text{SINR}_{\text{PU}}(k,i) = \text{SINR}_{\text{PU}_j}(k,i)$. In addition to this simple yet contrived simulation model, we evaluate our framework in more realistic settings: with additional developments to our channel model – including LoS/NLoS components, large and small-scale variations, and propagation environment specific parameters [28] – we perform optimized rate adaptation at both the PUs as well as the SU, i.e., $\arg\max_{R \geq 0} C$ via the bisection method [29]: these evaluations are described later in this section.

In Fig. 4 (L), we plot the Mean Square Error (MSE) of the model estimator (HMM EM) vs the number of iterations (i), where the MSE is evaluated as

$$\|\vec{\theta} - \hat{\vec{\theta}}^{(t)}\|_2^2 = \sum_{\theta \in \vec{\theta}} \mathbb{E}[(\theta - \hat{\theta}^{(t)})^2]. \tag{26}$$

We note that, with initial estimates of 0.5, i.e., p_{uv}=0.5, $\forall u, v \in \{0, 1\}$ and q_w=0.5, $w \in \{0, 1\}$, the MSE is decreased iteratively, as the estimation process goes through the E-step and the M-step in each iteration t until the estimator converges [37] to the true parameter vector $\vec{\theta}$ with an error/delta of η=10^{-8} ($\|\theta - \hat{\theta}^{(t)}\|^2 \leq 10^{-8}, \forall \theta \in \vec{\theta}$) in 45,000 iterations: this corresponds to an observation and estimation period of 135 s, considering a typical time-slot duration of 3 ms. The computational time complexity of this EM variant for HMMs is $O(\tau K \tilde{T})$ [37], where τ corresponds to the number of time-slots employed for observations, K refers to the number of channels in the discretized spectrum of interest, and $\tilde{T}$ refers to the number of iterations involved until MSE convergence – which depends on the consistency of channel occupancy measurements, driven by our observation model and the SU's sensing limitations.

On the same time-scale as the parameter estimation algorithm, focusing on the loss convergence of the PERSEUS algorithm with a discount factor of γ=0.9 and a termination threshold of

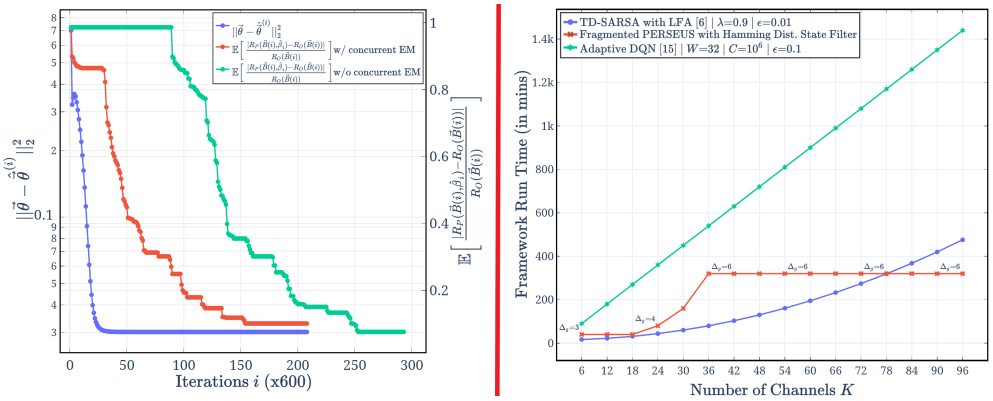

Fig. 4. The convergence of the MSE of the HMM EM algorithm to estimate $\vec{\theta}$, and the convergence of the loss of the fragmented PERSEUS algorithm with belief update simplification (L) | The computational time complexity bench-marking of our framework against comparable algorithms in the state-of-the-art (R)

ϵ=10^{-5}, wherein we define the expected loss as the difference between the utility obtained by the proposed PERSEUS framework, denoted by $R_P(\vec{B}(i))$ (discussed in Sec. II-D), and that obtained by an Oracle, which knows the exact occupancy behavior of the PUs in the network, denoted by $R_O(\vec{B}(i))$, we find that, as depicted in Fig. 4 (L), the loss convergence of PERSEUS is relatively slower while the parameter estimator is learning the transition model; as opposed to after the convergence of the parameter estimator, when we see a more consistent gradient towards the optimality. Also, note the normalized sub-optimality gap of 0.05, i.e., the average normalized difference between the utility obtained by our approximately optimal POMDP policy (post-convergence) and that of an Oracle (which knows the exact PU occupancy behavior throughout the simulation period) is 0.05 or 5%: this is a significant result because in spite of possessing no *a priori* knowledge of the underlying MDP's transition model, and operating in a noisy environment under sensing constraints, our approximate POMDP formulation solved with a randomized point-based value iteration scheme – with fragmentation and Hamming distance state filter heuristics – achieves a performance that is on par with an Oracle; note that an Oracle performs better than the *optimal* policy – so, solving for the *optimal* policy while sacrificing computational feasibility does not yield a tangible boost in spectrum white-space detection, thereby legitimizing the validity of our approach.

The computational time complexity of the fragmented PERSEUS algorithm is $O(\tilde{T}|\tilde{\mathcal{B}}|^2 2^{2\Delta_g})$ [34], where $\tilde{T}$ denotes the number of iterations constituting PERSEUS convergence, $|\tilde{\mathcal{B}}|$=$|\tilde{\mathcal{U}}|$ denotes the number of reachable beliefs obtained during the initial exploration phase, and Δ_g denotes the fragment size, i.e., the number of channels in the fragmented and discretized spectrum

of interest; note here that incorporating Hamming distance state filters to alleviate the computational intractability inherent in PERSEUS belief updates mitigates the doubly-exponential dependence on the fragment size. Moreover, Fig. 4 (L) depicts the computational time difference between running the parameter estimator and the PERSEUS algorithm concurrently via the iterative publisher-subscriber architecture, as opposed to initiating the PERSEUS run after the convergence of the parameter estimator: we cut down the time to completion of our HMM-POMDP framework by half by employing the former approach as opposed to the latter, without worsening the sub-optimality gap significantly. Finally, in Fig. 4 (R), we elucidate the results from the computational time complexity bench-marking of our fragmented PERSEUS with Hamming distance state filters scheme against TD-SARSA with LFA [6] and Adaptive DQN [15] on a $2\times$ 12-core Intel Xeon Gold 6126 @ 2.6 GHz compute node with 192 GB RAM [40]: as the number of channels in the discretized spectrum of interest increases – owing to fragmentation and belief update simplification heuristics – our solution scales better yielding a more computationally tractable performance relative to the other two.

In Fig. 5, we plot $\mathbb{P}(R(\vec{\phi}(i), \vec{B}(i)) \leq x)$ vs x, where x represents the reward value obtained by the PERSEUS agent in a time-slot, evaluated according to (14). We find that our framework obtains an average utility, i.e., $R(\vec{\phi}(i), \hat{\beta}_i)$ described in Sec. II-D, of 11.98 per time-step i, 125% higher than that achieved by the MEM with GC-CCE and MPE algorithm from [14], 96% higher than that achieved by the MEM with MEI-CCE and MPE algorithm from [14], and 42% more than that attained by the Neyman-Pearson Detector detailed above [11]. Compared to Imperfect HMM-MAP State Estimation ($=11.78$), our scheme achieves 2% higher utility ($=11.98$), thanks to an adaptive sensing strategy.

In Fig. 6 (L), we compare the performance of the proposed framework, denoted as "Fragmented PERSEUS with Belief Update Simplification," with the following state-of-the-art solutions detailed in current literature, in terms of the secondary network throughput achieved vis-à-vis PU interference:

- MEM with GC-CCE and MPE [14]: Minimum Entropy Merging (MEM) with Greedy Clustering based Channel Correlation Estimation (GC-CCE) and Markov Process Estimation (MPE), Correlation Threshold $\rho_{th}=0.7$, Number of clusters $T=6$, i.e., a channel sensing restriction of 6 – our solution offers a 104% improvement over this strategy;
- MEM with MEI-CCE and MPE [14]: Minimum Entropy Merging (MEM) with Minimum Entropy Increment Clustering based Channel Correlation Estimation (MEI-CCE) and

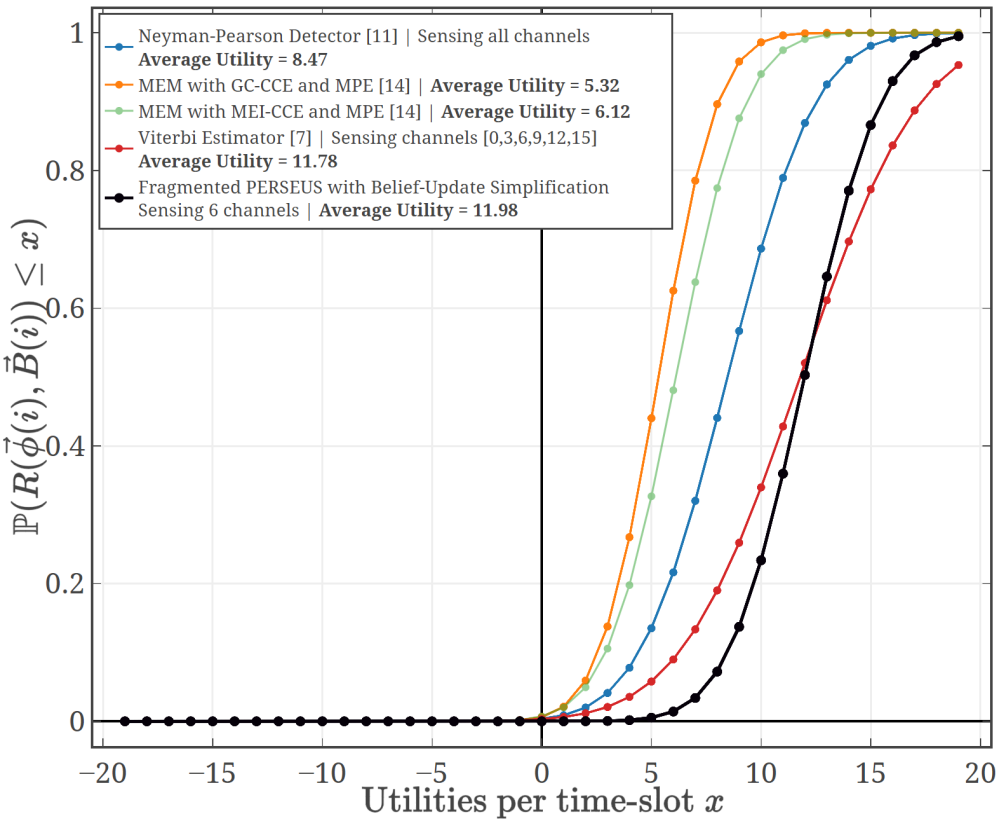

Fig. 5. The evaluation of the proposed solution, from an average utility per time-slot perspective, against a medley of approaches in the state-of-the-art: $\mathbb{P}(R(\vec{B}(i))\leq x)$ versus utility per time-slot x – where $R(\vec{\phi}(i), \vec{B}(i))$ is given by (13)

Markov Process Estimation (MPE), Correlation Threshold ρ_{th}=0.7, Number of clusters T=6, i.e., a channel sensing restriction of 6 – our solution achieves 38% better performance over this strategy;

- Imperfect HMM-MAP State Estimation [7]: The Viterbi algorithm, assuming *a priori* knowledge of the time-frequency Markovian correlation structure in PU occupancy behavior, with a channel sensing restriction of 6 – our solution attains a 6% boost over this strategy;
- Neyman-Pearson Detection [11]: A Neyman-Pearson Detector, assuming independence across channels and across time, with no channel sensing restrictions, an AND fusion rule across 300 samplings, and threshold determination via a false alarm probability of 30% – our solution offers a 25% enhancement over this strategy;
- Prior Perfect Model Knowledge + Fragmented PERSEUS with Belief Update Simplification: A Fragmented PERSEUS algorithm with Belief-Update Simplification (Hamming distance state filters), with prior occupancy behavior correlation model information – the proposed HMM EM + Fragmented PERSEUS with Hamming State Filters exhibits 3.75% worse

performance than this strategy, i.e., knowing the model beforehand offers a meagre 3.75% boost in performance compared to the proposed online concurrent model estimation and policy solver strategy - a testament to the accuracy of our estimator;

- Temporal Difference Learning via SARSA with Linear Function Approximation [6]: TD-SARSA with Linear Function Approximation (LFA) in single-agent deployment settings, with a sensing restriction of 6, a belief update heuristic constant λ=0.9, a discount factor of γ=0.9, a fixed exploration factor ϵ=0.01, and a raw false alarm probability of $p_{fa,1}$=5% – our solution exhibits a 3% superior performance over this strategy;
- Greedy Learning under Pre-Allocation [13]: Greedy Learning in single-agent deployment settings, with a channel sensing restriction of 6, and a time-varying exploration factor $\epsilon = \min(\frac{\beta}{i}, 1)$, where $\beta > \max(20, \frac{4}{\Delta^2_{\text{min}}})$, with Δ_{min} referring to the smallest Kullback-Liebler distance between a pair of channels – our solution offers a 10% enhancement over this strategy;
- g-statistics [13]: Learning with g-statistics and ACKs in single-agent deployment settings, with a channel sensing restriction of 6 – our solution achieves a 15% boost in performance over this strategy;
- Adaptive Deep Q-Networks [15]: An adaptive Deep Q-Network (DQN) with Experiential Replay (Memory Size C=10^6), 2048 input neurons, 4096 neurons with ReLU activation functions in each of the 2 hidden layers of the Neural Network, a Mean-Squared Error cost function with an Adam Optimizer, a Fixed Exploration Factor ϵ=0.1, a Learning Rate of α=10^{-4}, a Batch Size of W=32, and a sensing restriction of 6 – our solution offers a 9% improvement over this strategy.

Also, we note that our POMDP agent limits channel access when the penalty (λ) is high, leading to lower SU throughput and lower PU interference, and conversely, follows a more lenient channel access strategy when the penalty is low, resulting in higher SU throughput and higher PU interference. Generally speaking, Fig. 6 (L) depicts a trend of increasing SU throughput and increasing PU interference, as the penalty for missed detections, i.e., λ is lowered. Therefore, our framework provides a crucial practical tool in cognitive radio MAC design: the ability to tune the trade-off between the throughput obtained by the SU and the interference caused by it to PU transmissions in the network. Additionally, as evident in Fig. 6 (L), a variant of our framework with rate adaptation at both the PUs as well as the SU

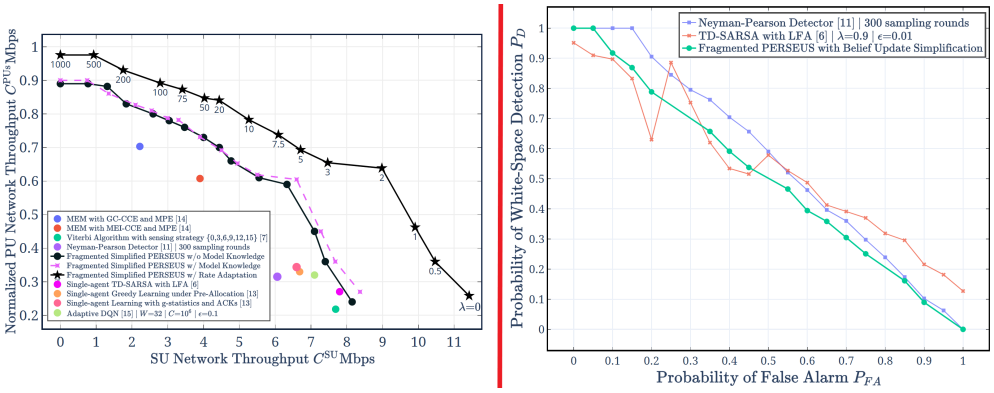

Fig. 6. The evaluation of SU and PU network throughputs for different values of λ with/without rate adaptation and with/without correlation model foreknowledge – along with comparisons with the state-of-the-art (L) | The Receiver Operating Characteristics (ROC) curve for our framework – along with those for comparable works in the state-of-the-art (R)

[28], [29] exhibits a significant enhancement in throughput performance. Finally, in Fig. 6 (R), we illustrate a False Alarm Probability (P_{FA}=$\mathbb{P}(\phi_k(i)$=$1|B_k(i)$=$0)$) v White-Space Detection Probability (P_D=$\mathbb{P}(\phi_k(i)$=$0|B_k(i)$=$0)$) curve for our fragmented PERSEUS with Hamming distance state filters framework – along with those for TD-SARSA with LFA [6] and Neyman-Pearson Detection [11]: compared to these two schemes in the state-of-the-art, our solution achieves better detection performances vis-à-vis false alarm constraints.

V. Multi-Agent Deployment Model: An Extension to the Single-Agent Setting

A. Distributed Multi-Agent Spectrum Sensing and Access

In this section, we evaluate the performance of the proposed framework: HMM EM + Fragmented PERSEUS with Belief Update Simplification, in distributed multi-agent deployment settings. Operating under the same signal and observation models as in Sec. II, consider a network of 3 PUs operating in an 18-channel radio environment, with their occupancy behaviors in this discretized spectrum of interest governed by Markovian time-frequency correlation structure ($\vec{\theta}$), and 12 SUs intelligently trying to access white-spaces in the spectrum (cooperatively [6] or opportunistically [13]), with an added restriction of being able to sense only 1 channel per SU per time-slot, as illustrated in Fig. 1.

The POMDP model described in Sec. II-D has been adapted to this multi-agent deployment setting by incorporating neighbor discovery, channel access rank allocation, and data aggregation algorithms into the original POMDP process flow, as depicted in Fig. 3. Designating the band-edges as the control channel, for neighbor discovery, each SU broadcasts its control frames (with a frame header and node identifier) over the control channel, and upon receiving control

messages from all its surrounding nodes, each SU checks if the expected RSSI of the radio signals corresponding to a certain node is above a threshold RSSI_{th}: if yes, adds that node's identifier to its list of neighbors.

With a similar control channel strategy for channel access rank allocation, we employ a quorum-based preferential ballot scheme to determine the order in which the *estimated-idle* channels are accessed by the SUs in the network. This procedure kicks in only after a quorum has been achieved, i.e., the number of neighbors identified by an SU should be equal to or exceed a node-specific pre-defined number. Over the control channel, each SU exchanges a ranked list of its neighbors in the decreasing order of their respective RSSIs, with itself being on the list at position-1 (ties are broken via uniform random choice). Upon receiving an *RSSI-ranked* list from one of its neighbors, each SU assigns points to each ranked position, with higher ranks getting larger point values, and re-broadcasts an *aggregated-ranked* list of neighbors (with itself being on the list) with the ranking based on the point-values aggregated across all the ranked lists received from its neighbors (ties are broken via uniform random choice). If the *aggregated-ranked* lists received from its neighbors matches the one at the SU, and this is true for a pre-specified consecutive period of time, a consensus has been reached, the channel access order is determined by this *harmonized-aggregated-ranked* list. If the *aggregated-ranked* lists received from its neighbors differ from the one at the SU, then the SU repeats the re-ranking of these list members based on their new aggregated point-values and broadcasts the new *aggregated-ranked* list to its neighbors over the control channel. This repetitive process continues until a consensus is reached.

Analyzing the performance of the proposed framework (HMM EM + Fragmented PERSEUS with Belief-Update Simplification) against other distributed multi-agent schemes in the state-of-the-art, as shown in Fig. 7, we find that our framework, in terms of the average utility $R(\vec{\phi}(i), \hat{\beta}(i))$ obtained per time-slot, out-performs the distributed, cooperative, ϵ-greedy TD-SARSA with Linear Function Approximation framework from [6] by 43%; out-performs the distributed, cooperative, time-decaying ϵ-greedy algorithm with channel access rank pre-allocations from [13] by 84%; and out-performs the distributed, opportunistic, g-statistics algorithm with ACKs (without channel access rank pre-allocations) from [13] by 324%. The computational time complexity of the RSSI thresholding scheme for neighbor discovery is $O(\tilde{J}^2)$ [13], where $\tilde{J}$ corresponds to the number of cognitive radios in the deployment of interest; while that of the quorum-based preferential ballot scheme for channel access rank allocation is $O(\tilde{T}\tilde{J}^2)$ [13],

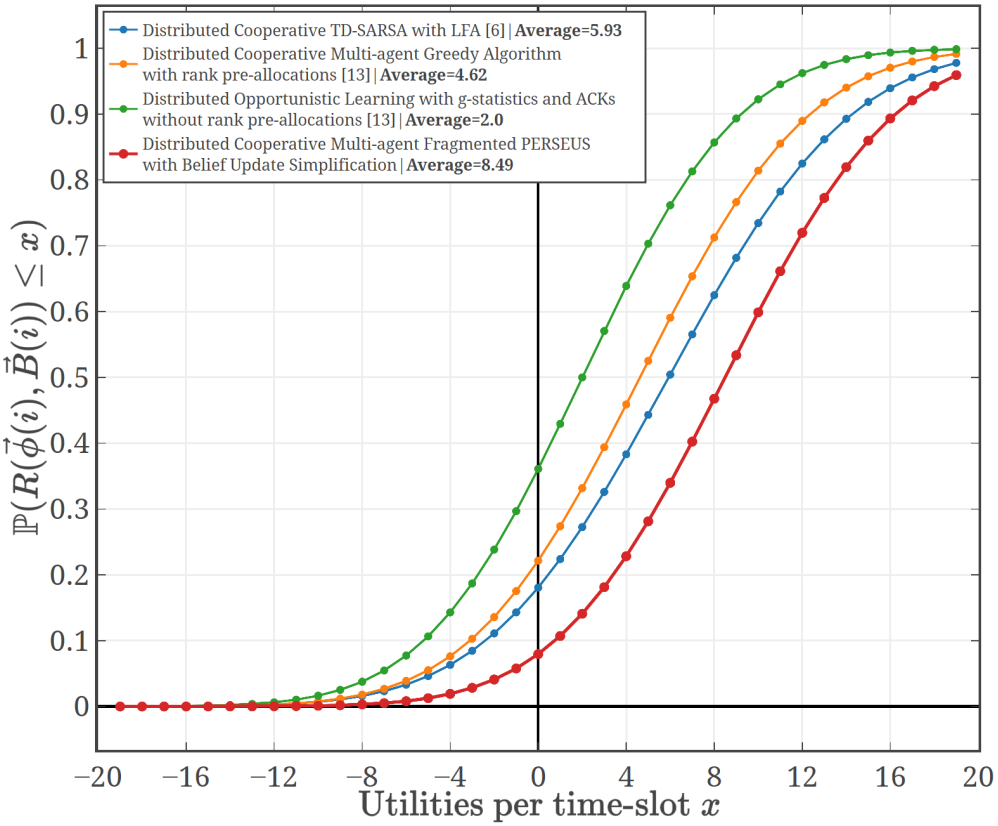

Fig. 7. An evaluation of the performance (average utility per time-slot) of the proposed framework in a distributed multi-agent deployment setting, against other distributed cooperative/opportunistic multi-agent channel sensing & access frameworks in the state-of-the-art: $\mathbb{P}(R(\vec{B}(i)) \leq x)$ versus utility per time-slot x – where $R(\vec{\phi}(i), \vec{B}(i))$ is given by (13)

where $\tilde{T}$ corresponds to the number of iterations involved until a consensus is reached – which depends on the mobility patterns of these nodes along with the temporal evolution of the peer-to-peer link qualities.

B. Centralized Multi-Agent Spectrum Sensing and Access: SC2 Active Incumbent Emulation

In order to evaluate the performance of the proposed framework (HMM EM + Fragmented PERSEUS with Belief Update Simplification) in real-world settings, we retrofit it into the MAC layer (channel & bandwidth allocation) of our BAM! Wireless radio [19], and analyze its operational capabilities in the DARPA SC2 Active Incumbent scenario [20] emulated on the Colosseum [41], [42]. The DARPA SC2 Active Incumbent scenario consists of a Terminal Doppler Weather Radar (TDWR) system functioning as the PU, and 5 competitor networks (ours included), each constituting 2 UNII WLANs: 2 Access Points (APs) and 4 STAtions (STAs) per AP, serving as the SUs, in a 10 MHz radio environment (995 MHz to 1005 MHz), for 330 seconds of emulation on the Colosseum [20]. During the Active Incumbent scenario emulation, every

competitor network receives network flows from the Colosseum which need to be delivered to the appropriate destination nodes within the network, while satisfying the imposed QoS mandates per flow (for example: max_latency, min_throughput, file_transfer_deadline, etc.). If the QoS mandates imposed on a particular network flow have been satisfied for a pre-specified period of time (referred to as *Measurement Periods* (MPs)), then the Individual Mandates (IMs) associated with the flow are said to have been met. With this concept of IMs in mind, we can define the points achieved or the *score* of a participant network corresponding to a certain time-slot i as $\sum_{v \in \mathcal{V}_i} \mathrm{p}_v$, where $\mathcal{V}_i$ denotes the set of IMs achieved by a participant network in time-slot i. The scenario also incorporates ensemble performance thresholds, i.e., all the participant networks should meet the scoring threshold of 8 [20]: if a participant network fails to meet this threshold, all the participant networks get the lowest score, i.e., the score corresponding to that achieved by this under-performing network, else, if all the participant networks in the emulation achieve scores that exceed the threshold, their scores are incremented beyond this threshold commensurate with the IMs achieved by them in that time-slot.

After having understood the scoring mechanism involved in the DARPA SC2, we can now evaluate the performance of the proposed framework retrofitted into our standard BAM! Wireless radio [19] against other radios designed by our peers who also participated in this competition, in addition to a performance comparison with the weighted PSD + CIL [21] channel & bandwidth allocation scheme employed, as a standard out-of-the-box protocol, in our traditional BAM! Wireless network. Leveraging the aggregated PSD measurements obtained at the gateway node of our BAM! Wireless network, as shown in Fig. 2 (R), we evaluate the scores of the proposed framework retrofitted into our standard BAM! Wireless radios against our traditional channel & bandwidth allocation scheme (titled "Standard BAM! Wireless Radio [Purdue]), and against the designs of our peers (identified by their collaboration network registered IP address [21], "172.30.210.191 [Peer]" and "172.30.210.181 [Peer]"): in terms of the average score achieved per time-slot, we deduce from Fig. 8 that the proposed framework ("BAM! Wireless Radio + HMM EM + Fragmented PERSEUS with Belief Update Simplification") out-performs our traditional channel & bandwidth allocation scheme (a simple weighted PSD + CIL heuristic) by 21%; provides a 56% better performance than one of our peers, identified by "172.30.210.181"; and attains an 81% boost in performance over another one of our peers, identified by "172.30.210.191".

To evaluate the proposed neighbor discovery (RSSI thresholding) and channel access rank allocation (quorum-based preferential ballot) heuristics from a computational time complexity

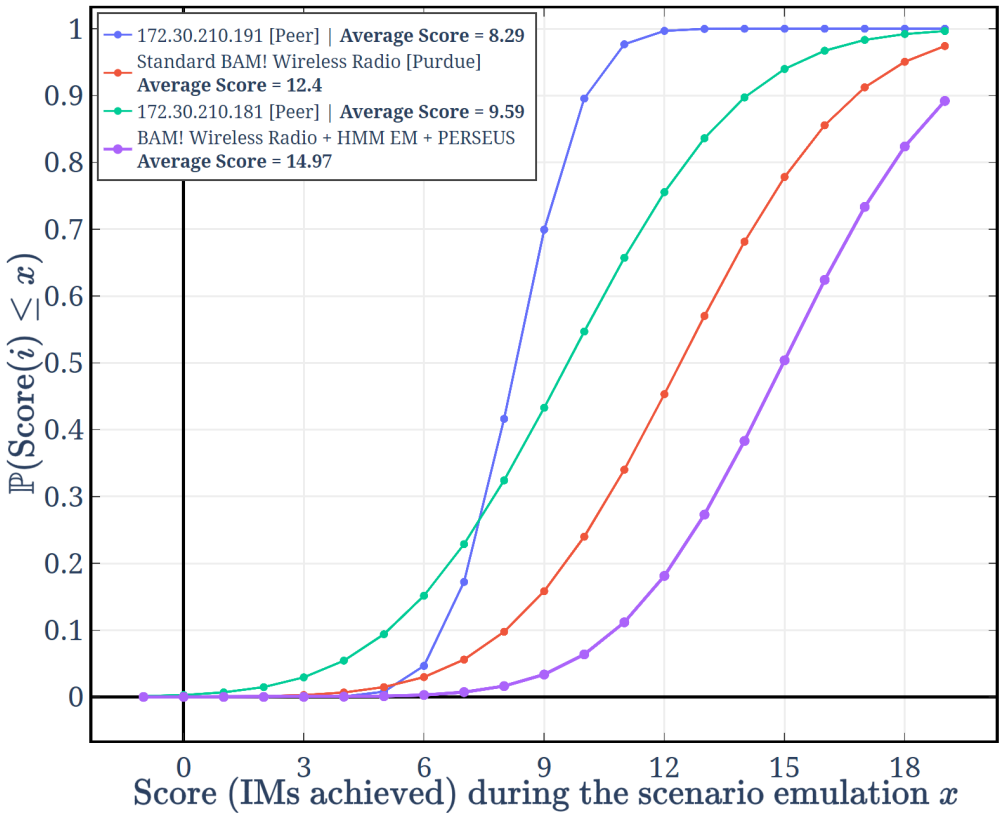

Fig. 8. An evaluation of the performance (scores/Individual Mandates (IMs) achieved) of our solution by retrofitting the proposed POMDP framework into our BAM! Wireless cognitive radio network design, with respect to an emulation of the DARPA SC2 Active Incumbent scenario, against other competitor network radio designs: $\mathbb{P}(\text{Score} \leq x)$ versus the scores achieved during the course of this emulation x

perspective, we retrofit these schemes into the control channel design, collaboration, and data aggregation modules of the Purdue BAM! Wireless radio [19], and analyse their feasibility in emulations of highly mobile real-world scenarios – namely, military deployments in the Alleys of Austin scenario [27] (urban: $5\times$ [9−guardsmen + 1−UAV]) and disaster relief deployments in the Payline scenario [26] (urban: $5\times$ [9−EMTs + 1−HQ]). Along with the scenario-specific node mobility patterns, the results of these emulations are shown in Fig. 9: neighbor discovery list changes are minimal in spite of node mobility (with an RSSI threshold of 22 dB), and distributed convergence of channel access rank allocation across the ensemble is achieved within the first few iterations in any given 10 s time-step – more specifically, for the Alleys of Austin scenario, channel access rank list convergence among the ensemble nodes is illustrated for the 10 s time-step between 2019-09-03 | 01:37:22 to 01:37:32; and between 2019-09-03 | 03:42:40 to 03:42:50 for the Payline scenario.

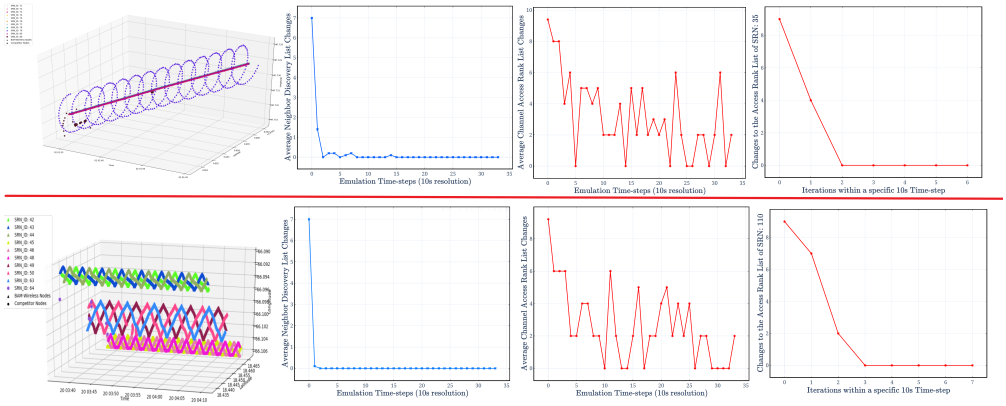

Fig. 9. The evaluation of our neighbor discovery and channel access rank allocation schemes in the DARPA SC2 Alleys-of-Austin (top) and Payline (bottom) scenarios: the mobility patterns of the constituent nodes (a); the average number of neighbor discovery list changes (b) and the average number of channel access rank list changes (c) across the entire emulation period; and the convergence visualization of channel access rank allocation in a specific 10s time-step at a specific cognitive radio in the network (d)

C. Implementation Feasibility of the Multi-Agent POMDP Model on an ESP32 WLAN Testbed

We employ 8 ESP32 radios [17], with each one embedded in a GCTronic e-puck2 robot [18], categorized into a network of 3 PUs (and their 3 corresponding sinks) occupying 6 channels in the discretized spectrum of interest according to a Markovian time-frequency correlation structure (described by (6)), and 2 independent SUs, with each having the capability of sensing only one channel at a time, intelligently trying to exploit the white-spaces in the spectrum. The detailed methodology of this implementation is provided below:

- Considering a network with $J{=}3$ PUs and one SU (work split over 2 ESP32 radios due to design limitations) with a channel sensing restriction of $\kappa{=}2$ out of $K{=}6$ channels in the discretized spectrum of interest, and assuming a linear AWGN observation model, with a Rayleigh channel fading model (discussed in Sec. II-A), we simulate the occupancy behavior of the PUs according to a Markovian time-frequency correlation structure parameterized by $\vec{\theta}{=}[\vec{p}, \vec{q}]^\intercal$, where $\vec{p}{=}[p_{00}{=}0.1, p_{01}{=}0.3, p_{10}{=}0.3, p_{11}{=}0.7]^\intercal$ and $\vec{q}{=}[q_0{=}0.3, q_1{=}0.8]^\intercal$; and solve for the spectrum sensing and access policy using PERSEUS, embedded with a concurrent parameter estimation algorithm learning the parameter vector $\vec{\theta}$, by mimicking the observational capabilities of the actual ESP32 radios. Note this step is performed on a PC.
- The simulated PU occupancy behavior, Markovian correlated according to (6) and parameterized by $\vec{\theta}$, and the time-slot specific channel access decisions (derived off of the POMDP approximately optimal sensing policy and the simulated PU occupancy behavior), are stored

in databases (for export onto the ESP32 network).

- Peer-to-Peer communication links are established between a PU ESP32 radio and its sink, using the 3 ESP32 radios designated as PUs. In other words, 3 wireless communication links are established: one for each ESP32 PU pair (a source and a sink), over WiFi (2.4 GHz) and using a channel according to the occupancy information detailed in the exported PU occupancy database, in time-slot i.
- Note here that in this ESP32 PU network implementation, in time-slot i, while establishing a wireless communication link between a ESP32 PU $j \in \{1, 2, 3\}$ and its respective sink $i \in \{1, 2, 3\}$ s.t. i is the designated sink for PU j, i.e., while forming link l_{ij} over channel $k_{l_{ij}} = k \in \{1, 2, \ldots, 6\}$ (as determined by the exported PU occupancy database which contains simulated PU occupancy behavior according to the Markovian time-frequency correlation structure described above) such that $k_{l_{ij}} \neq k_{l_{i',j'}}, \forall i, i' \in \{1, 2, 3\},\ j, j' \in \{1, 2, 3\}$, PU j serves as an Access Point (AP) accepting transmission requests from PU i, which is designated as a STAtion (STA). In the next synchronized time-slot $i + 1$, this link l_{ij} moves to channel $k' \in \{1, 2, \ldots, 6\}$, as detailed in the exported PU occupancy database. This same procedure takes place for the other two PU communication links in every time-slot until the end of the implementation evaluation period.
- Although the PC-based POMDP solver employs an SU which can access 2 channels at a time in order to deliver its flows (see the access part of the POMDP formulation in Sec. II-D), we employ 2 ESP32 SU radios in the network (serving as one), with the channel access work synchronously and evenly split between the two, due to the actual physical design limitations of the ESP32 radio that it can only access one channel at a time, forcing us to be creative: split the 2 channel access decision in time-slot i, as determined by the time-slot specific POMDP channel access database, into a 1 channel access action at each ESP32 SU radio. Next, based on whether the channel access at the 2 ESP32 SU radios was successful, we compute the success rate.

The channel access success rate metric defined as

$$\text{Channel Access Success Probability} = \frac{\sum_{j=1}^{2} \mathcal{I}\left\{B_{k_{SU_j}}(i) = 0\right\}}{2}, \tag{27}$$

where $\mathcal{I}$ corresponding to $\mathcal{I}\left\{B_{k_{SU_j}}(i) = 0\right\}$ is an indicator variable whose value is 1 if the channel accessed by the ESP32 SU $j \in \{1, 2\}$ in time-slot i is not occupied by a PU ESP32 radio, and $B_{k_{SU_j}} \in \{0, 1\}$ is the occupancy variable of the channel accessed by the ESP32 SU j

in time-slot i, is evaluated per time-slot i, and the resultant channel access success probability is 95.75%.

VI. Conclusion

In this paper, we formulate the spectrum sensing and access problem in resource-constrained radio ecosystems as an approximate POMDP, which leverages learning of the PU occupancy correlation model via the Baum-Welch algorithm and solving for an approximately optimal sensing and access policy via the PERSEUS algorithm – with fragmentation and Hamming distance state filter heuristics to enforce computational tractability. Through system simulations, we demonstrate the advantages of exploiting the correlation structure – as opposed to Neyman-Pearson Detector which assumes independence – and of adapting the spectrum sensing decision to optimize the performance – as opposed to Viterbi, which uses a fixed sensing strategy. We also demonstrate the feasibility of a concurrent learning and decision-making framework, as opposed to state-of-the-art correlation-coefficient based clustering algorithms, which rely on pre-loaded datasets for determining the correlation in the PU occupancy behavior. Our framework enables a critical feature in practical scenarios: the ability of the SU to regulate the interference caused to PUs, by adjusting a penalty parameter. Also, extending our single-agent model to multi-agent settings, we demonstrate superior performance over the state-of-the-art, in both centralized and distributed deployment settings (collaborative and opportunistic access).

References

[1] B. Keshavamurthy and N. Michelusi, "Learning-based Cognitive Radio Access via Randomized Point-Based Approximate POMDPs," 2020, Under review at IEEE ICC 2021.

[2] Ericsson, "5G use cases–Explore how 5G will revolutionize 5 key industries including: TV and media; manufacturing; healthcare; telecommunications; and transportation and infrastructure." *Ericsson*, 2019. [Online]. Available: https://www.ericsson.com/en/5g/use-cases

[3] D. Goldin, "Keep 5G Safe From Chinese Domination," *The Wall Street Journal*, 2020. [Online]. Available: https://www.wsj.com/articles/keep-5g-safe-from-chinese-domination-11580342112

[4] S. Maleki, S. P. Chepuri, and G. Leus, "Energy and throughput efficient strategies for cooperative spectrum sensing in cognitive radios," in *2011 IEEE 12th International Workshop on Signal Processing Advances in Wireless Communications*, June 2011, pp. 71–75.

[5] K. Cohen, Q. Zhao, and A. Scaglione, "Restless Multi-Armed Bandits under time-varying activation constraints for dynamic spectrum access," in *2014 48th Asilomar Conference on Signals, Systems and Computers*, Nov 2014, pp. 1575–1578.

[6] J. Lundén, S. R. Kulkarni, V. Koivunen, and H. V. Poor, "Multiagent Reinforcement Learning Based Spectrum Sensing Policies for Cognitive Radio Networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 5, pp. 858–868, Oct 2013.

[7] C. Park, S. Kim, S. Lim, and M. Song, "HMM Based Channel Status Predictor for Cognitive Radio," in *2007 Asia-Pacific Microwave Conference*, Dec 2007, pp. 1–4.

[8] L. Ferrari, Q. Zhao, and A. Scaglione, "Utility Maximizing Sequential Sensing Over a Finite Horizon," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3430–3445, July 2017.

[9] N. Michelusi and U. Mitra, "Cross-Layer Estimation and Control for Cognitive Radio: Exploiting Sparse Network Dynamics," *IEEE Transactions on Cognitive Communications and Networking*, vol. 1, no. 1, pp. 128–145, March 2015.

[10] N. Michelusi, M. Nokleby, U. Mitra, and R. Calderbank, "Multi-Scale Spectrum Sensing in Dense Multi-Cell Cognitive Networks," *IEEE Transactions on Communications*, vol. 67, no. 4, pp. 2673–2688, April 2019.

[11] S. Mosleh, A. A. Tadaion, and M. Derakhtian, "Performance analysis of the Neyman-Pearson fusion center for spectrum sensing in a Cognitive Radio network," in *IEEE EUROCON 2009*, May 2009, pp. 1420–1425.

[12] S. Yin, D. Chen, Q. Zhang, M. Liu, and S. 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.

[13] A. Anandkumar, N. Michael, and A. Tang, "Opportunistic Spectrum Access with Multiple Users: Learning under Competition," in *Proceedings of the 29th Conference on Information Communications*, ser. INFOCOM'10. IEEE Press, 2010, p. 803–811.

[14] M. Gao, X. Yan, Y. Zhang, C. Liu, Y. Zhang, and Z. Feng, "Fast Spectrum Sensing: A Combination of Channel Correlation and Markov Model," in *2014 IEEE Military Communications Conference*, Oct 2014, pp. 405–410.

[15] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep Reinforcement Learning for Dynamic Multichannel Access in Wireless Networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 257–265, 2018.

[16] M. Rosker, "Spectrum Collaboration Challenge (SC2)," *Defense Advanced Research Projects Agency (DARPA) Spectrum Collaboration Challenge (SC2)*, 2018. [Online]. Available: https://www.darpa.mil/program/spectrum-collaboration-challenge

[17] Espressif Systems (Shanghai) Co. Ltd., "Espressif ESP32: A Different IoT Power and Performance," *Espressif*, 2019. [Online]. Available: https://www.espressif.com/en/products/hardware/esp32/overview

[18] GCTronic, "Epuck2 Specifications and General Wiki," *GCTronic e-puck2 online wiki*, 2020. [Online]. Available: https://www.gctronic.com/doc/index.php/e-puck2

[19] DARPA, "Purdue University BAM! Wireless Radio," *Defense Advanced Research Projects Agency (DARPA) Spectrum Collaboration Challenge (SC2)*, 2019. [Online]. Available: https://archive.darpa.mil/sc2/news/spectrum-collaboration-challenge-awards-four-teams-with-half-prizes

[20] DARPA-SC2-Freshdesk, "Active Incumbent Scenario Specifications," *DARPA Spectrum Collaboration Challenge (SC2)*, 2019. [Online]. Available: https://sc2colosseum.freshdesk.com/support/solutions/articles/22000239489-active-incumbent-

[21] DARPA SC2 GitLab CIL Schematics, "CIL Specifications," *DARPA Spectrum Collaboration Challenge (SC2)*, 2019. [Online]. Available: https://gitlab.com/darpa-sc2-phase3/CIL

[22] F. A. P. d. Figueiredo, D. Stojadinovic, P. Maddala, R. Mennes, I. Jabandžić, X. Jiao, and I. Moerman, "SCATTER PHY: A Physical Layer for the DARPA Spectrum Collaboration Challenge," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2019, pp. 1–6.

[23] R. J. Baxley and R. S. Thompson, "Team Zylinium DARPA Spectrum Collaboration Challenge Radio Design and Implementation," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2019, pp. 1–6.

[24] D. Stojadinovic, F. A. P. de Figueiredo, P. Maddala, I. Seskar, and W. Trappe, "SC2 CIL: Evaluating the Spectrum Voxel

Announcement Benefits," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2019, pp. 1–6.

[25] S. Giannoulis, C. Donato, R. Mennes, F. A. P. de Figueiredo, I. Jabandžic, Y. De Bock, M. Camelo, J. Struye, P. Maddala, M. Mehari, A. Shahid, D. Stojadinovic, M. Claeys, F. Mahfoudhi, W. Liu, I. Seskar, S. Latre, and I. Moerman, "Dynamic and Collaborative Spectrum Sharing: The SCATTER Approach," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2019, pp. 1–6.

[26] D.-S.-F. PE2, "Payline (2 Stage)," *DARPA Spectrum Collaboration Challenge (SC2)*, 2019. [Online]. Available: https://sc2colosseum.freshdesk.com/support/solutions/articles/22000234543-payline-2-stage-7065-

[27] D.-S.-F. CE, "Alleys of Austin with Points - 5 Teams," *DARPA Spectrum Collaboration Challenge (SC2)*, 2019. [Online]. Available: https://sc2colosseum.freshdesk.com/support/solutions/articles/22000237879-pe2-alleys-of-austin-w-points-5-team-7013-

[28] D. Tse and P. Viswanath, "Fundamentals of wireless communication," 2005.

[29] Y. Sun, Baricz, and S. Zhou, "On the monotonicity, log-concavity, and tight bounds of the generalized marcum and nuttall q-functions," *IEEE Transactions on Information Theory*, vol. 56, no. 3, pp. 1166–1186, 2010.

[30] R. I. C. Chiang, G. B. Rowe, and K. W. Sowerby, "A Quantitative Analysis of Spectral Occupancy Measurements for Cognitive Radio," in *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, April 2007, pp. 3016–3020.

[31] M. A. McHenry, P. A. Tenhula, D. McCloskey, D. A. Roberson, and C. S. Hood, "Chicago Spectrum Occupancy Measurements & Analysis and a Long-term Studies Proposal," in *Proceedings of the First International Workshop on Technology and Policy for Accessing Spectrum*, ser. TAPAS '06. New York, NY, USA: ACM, 2006. [Online]. Available: http://doi.acm.org/10.1145/1234388.1234389

[32] Q. Jiang, M. Hlynka, P. H. Brill, and C. H. Cheung, "Reversibility checking for markov chains," 2018.

[33] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and Acting in Partially Observable Stochastic Domains," *Artif. Intell.*, vol. 101, no. 1–2, p. 99–134, May 1998.

[34] M. T. J. Spaan and N. A. Vlassis, "Perseus: Randomized Point-based Value Iteration for POMDPs," *CoRR*, vol. abs/1109.2145, 2011. [Online]. Available: http://arxiv.org/abs/1109.2145

[35] S. Giannoulis, C. Donato, R. Mennes, F. A. P. de Figueiredo, I. Jabandžic, Y. De Bock, M. Camelo, J. Struye, P. Maddala, M. Mehari, A. Shahid, D. Stojadinovic, M. Claeys, F. Mahfoudhi, W. Liu, I. Seskar, S. Latre, and I. Moerman, "Dynamic and Collaborative Spectrum Sharing: The SCATTER Approach," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2019, pp. 1–6.

[36] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *Ann. Math. Statist.*, vol. 37, no. 6, pp. 1554–1563, 12 1966. [Online]. Available: https://doi.org/10.1214/aoms/1177699147

[37] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE 77, no. 2*, pp. 257–286, Feb 1989.

[38] J. Pineau, G. Gordon, and S. Thrun, "Point-Based Value Iteration: An Anytime Algorithm for POMDPs," in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, ser. IJCAI'03. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, p. 1025–1030.

[39] N. L. Zhang and W. Zhang, "Speeding Up the Convergence of Value Iteration in Partially Observable Markov Decision Processes," *Journal of Artificial Intelligence Research*, vol. 14, p. 29–51, Feb 2001. [Online]. Available: http://dx.doi.org/10.1613/jair.761

[40] D. Duplyakin, R. Ricci, A. Maricq, G. Wong, J. Duerig, E. Eide, L. Stoller, M. Hibler, D. Johnson, K. Webb, A. Akella, K. Wang, G. Ricart, L. Landweber, C. Elliott, M. Zink, E. Cecchet, S. Kar, and P. Mishra, "The design and operation of

CloudLab," in *Proceedings of the USENIX Annual Technical Conference (ATC)*, Jul. 2019, pp. 1–14. [Online]. Available: https://www.flux.utah.edu/paper/duplyakin-atc19

[41] DARPA, "Radio Command and Control API," *Defense Advanced Research Projects Agency (DARPA) Spectrum Collaboration Challenge (SC2)*, 2018. [Online]. Available: https://sc2colosseum.freshdesk.com/support/solutions/articles/22000220460-radio-command-and-control-c2-api

[42] DARPA-SC2-Freshdesk, "Scenarios Summary List," *Defense Advanced Research Projects Agency (DARPA) Spectrum Collaboration Challenge (SC2)*, 2019. [Online]. Available: https://sc2colosseum.freshdesk.com/support/solutions/articles/22000236679--scenarios-summary-list-phase-3-