Utility Maximization in Cognitive Radio Networks using POMDP Approximate Value Iteration methods

Bharath Keshavamurthy, Nicolò Michelusi

[NM: title is a bit too long..] *Abstract—*

*Index Terms—***Hidden Markov Model, POMDP, and the PERSEUS Algorithm**

I. INTRODUCTION

II. SYSTEM MODEL

[NM: add figure with system model]

A. Signal Model

We consider a network consisting of P licensed users termed the Primary Users (PUs) and one cognitive radio node termed the Secondary User (SU) equipped with a spectrum sensor. The objective of the SU is to opportunistically access portions of the spectrum left unused by the PUs in order to maximize its own throughput. To this end, the SU should learn how to intelligently access spectrum holes (white-spaces) intending to maximize its throughput while maintaining strict non-interference compliance with incumbent transmissions. The wideband signal received at the SU receiver at time n is denoted as $y(n)$ and is given by

$$y(n) = \sum_{p=1}^{P} \sum_{l=0}^{L_p-1} h_p(l) x_p(n-l) + v(n), \qquad (1)$$

where $y(n)$ is expressed as a convolution of the signal $x_p(n)$ of the pth PU with the channel impulse response $h_p(n)$, and $v(n)$ denotes additive white Gaussian noise (AWGN) with variances σ_v^2. Eq. (1) can be written in the frequency domain by taking a K-point DFT which decomposes the observed wideband signal into K discrete narrow-band components as

$$Y_k(i) = \sum_{p=1}^{P} H_{p,k}(i) X_{p,k}(i) + V_k(i), \qquad (2)$$

where $i \in \{1, 2, 3, \ldots, T\}$ represents the time index; $k \in \{1, 2, 3, \ldots, K\}$ represents the index of the components in the frequency domain; $V_k(i) \sim \mathcal{CN}(0, \sigma_V^2)$ represents a circularly symmetric additive complex Gaussian noise sample, i.i.d across channel indices and across time indices; $X_{p,k}(i)$ is the signal of the pth PU in the frequency domain, and $H_{p,k}(i)$ is its frequency domain channel. The noise samples are assumed to be independent of the occupancy state of the channels. We further assume that the P PUs employ an orthogonal access to the spectrum (e.g., OFDMA) so that

$X_{p,k}(i) X_{q,k}(i) = 0, \ \forall p \neq q$. Thus, letting p_k be the index of the PU that contributes to the signal in the kth spectrum band (possibly, $p_k = 0$ if no PU is transmitting in the kth spectrum band), and letting $H_k(i) = H_{p_k,k}(i)$ and $X_k(i) = X_{p_k,k}(i)$, we can rewrite (2) as

$$Y_k(i) = H_k(i) X_k(i) + V_k(i). \qquad (3)$$

Thus, $H_k(i)$ represents the kth DFT coefficient of the impulse response $h_{p_k,k}(n)$ of the channel in between the PU operating on the kth spectrum band and the SU, at time i; we model it as a zero-mean circularly symmetric complex Gaussian random variable with variance σ_H^2, $H_k \sim \mathcal{CN}(0, \sigma_H^2)$, i.i.d. across frequency bands, over time, and independent of the occupancy state of the channels.

B. PU Spectrum Occupancy Model

We now introduce the model of PU occupancy over time and across the frequency domain. We model each $X_k(i)$ as

$$X_k(i) = \sqrt{P_{tx}} B_k(i) S_k(i), \qquad (4)$$

where P_{tx} is the transmission power of the PUs, $S_k(i)$ is the transmitted symbol modelled as a constant amplitude signal, $|S_k(i)| = 1$, i.i.d. over time and across frequency bands; [1] $B_k(i) \in \{0, 1\}$ is the binary spectrum occupancy variable, with $B_k(i) = 1$ if the kth spectrum band is occupied by a PU at time i, and $B_k(i) = 0$ otherwise. Therefore, the PU occupancy behavior in the entire wideband spectrum of interest at time i, discretized into narrow-band frequency components can be modeled as the vector

$$\vec{B}(i) = [B_1(i), B_2(i), B_3(i), \cdots, B_K(i)]^T \in \{0, 1\}^K. \quad (5)$$

PUs join and leave the spectrum at random times. To capture this temporal correlation in the spectrum occupancy dynamics of PUs, we model the spectrum occupancy dynamics as a Markov process: given $\vec{B}(i)$, the spectrum occupancy state at time index i, $\vec{B}(i+1)$ is independent of the past, $\vec{B}(j)$, $j < i$; $j, i \in \{1, 2, 3, \ldots, T\}$, i.e.

$$\mathbb{P}(\vec{B}(i+1)|\vec{B}(j), \ \forall j \leq i) = \mathbb{P}(\vec{B}(i+1)|\vec{B}(i)). \quad (6)$$

Additionally, when joining the spectrum pool, PUs occupy a number of adjacent spectrum bands, and may vary their

---

[1]In the case where $S_k(i)$ does not have constant amplitude, we may approximate $H_k(i) S_k(i)$ as complex Gaussian with zero mean and variance $\sigma_H^2 \mathbb{E}[|S_k(i)|^2]$, without any modification to the subsequent analysis.

spectrum needs depending on traffic demands, channel conditions, etc. To capture this behavior, we model $\vec{B}(i)$ as having Markovian correlation across the bands as,

$$\mathbb{P}(\vec{B}(i+1)|\vec{B}(i)) \tag{7}$$
$$= \mathbb{P}(B_1(i+1)|B_1(i)) \prod_{k=2}^{K} \mathbb{P}(B_k(i+1)|B_k(i), B_{k-1}(i+1)).$$

That is, the spectrum occupancy at time $i+1$ in frequency band k, $B_k(i+1)$, depends on the occupancy state of the adjacent spectrum band at the same time, $B_{k-1}(i+1)$, and that of the same spectrum band k in the previous time index i, $B_k(i)$.

C. Spectrum Sensing Model

In order to detect the available spectrum holes, the SU performs spectrum sensing. However, owing to physical design limitations at the SU's spectrum sensor **[NM: citation?]**, not all channels in the discretized spectrum can be sensed. Therefore, due to limited sensing capabilities, the SU can sense only κ out of K spectrum bands at any given time, with $1 \leq \kappa \leq K$. Let $\mathcal{K}_i \subseteq \{1, 2, \ldots, K\}$ with $|\mathcal{K}_i| \leq \kappa$ be the set of indices corresponding to the spectrum bands sensed by the SU at time i, which is part of our design. Then, we model the emission process of the HMM as

$$\vec{Y}(i) = [Y_k(i)]_{k \in \mathcal{K}_i}, \tag{8}$$

where $Y_k(i)$ is given by (3). The true states $\vec{B}(i)$ encapsulate the actual occupancy behavior of the PU and the measurements at the SU are noisy observations of these true states which are modeled to be the observed states of a Hidden Markov Model (HMM). Given the spectrum occupancy vector $\vec{B}(i)$ and the set of sensed spectrum bands $\mathcal{K}_i$, the probability density function of $\vec{Y}(i)$ is expressed as

$$f(\vec{Y}(i)|\vec{B}(i), \mathcal{K}_i) = \prod_{k \in \mathcal{K}_i} f(Y_k(i)|B_k(i)), \tag{9}$$

owing to the independence of channels, noise, and transmitted symbols across frequency bands. **[NM: it should not be too difficult to incorporate channel correlation across frequency bands]**: [BK: What does this mean? Given the states, the observations are independent across channels. So, why do we need to talk about channel correlation here?] Moreover,

$$Y_k(i)|B_k(i) \sim \mathcal{CN}(0, \sigma_H^2 P_{tx} B_k(i) + \sigma_V^2). \tag{10}$$

Now, we model the spectrum access scheme of the SU as a Partially Observable Markov Decision Process (POMDP) wherein the goal of the POMDP agent is to devise an optimal sensing and access policy in order to maximize its throughput while maintaining strict non-interference compliance with incumbent transmissions.

D. POMDP Agent Model

The agent's limited sensing capabilities coupled with its noisy observations result in an increased level of uncertainty at the agent's end about the occupancy state of the spectrum under consideration and the exact effect of executing an action on the radio environment. The transition model of the underlying MDP as described by (7), is denoted by A and is learnt by the agent by interacting with the radio environment. The emission model is denoted by M and is given by (9), with $f(Y_k(i)|B_k(i))$ given by (10). We model the POMDP as a tuple $(\mathcal{B}, \mathcal{A}, \mathcal{Y}, A, M)$ where $\mathcal{B} \equiv \{0, 1\}^K$ represents the state space of the underlying MDP with states $\vec{B}$ given by all possible realizations of the spectrum occupancy vector as described by (3), $\mathcal{A}$ represents the action space of the agent, given by all $\binom{K}{\kappa}$ possible combinations in which the κ spectrum bands are chosen to be sensed out of K at any given time; and $\mathcal{Y}$ represents the observation space of the agent based on the signal model outlined in the previous subsection. At the beginning of each time index i, the agent selects κ spectrum bands out of K, thus defining the sensing set $\mathcal{K}_i$, performs spectrum sensing on these spectrum bands, observes $\vec{Y}(i) \in \mathcal{Y}$, and updates its belief of the current spectrum occupancy $\vec{B}(i)$ as

$$b_i = b_{\mathcal{K}_i}^{\vec{Y}(i)}(\vec{B}') = \mathbb{P}(\vec{B}(i) = \vec{B}'|\vec{Y}(i), \mathcal{K}_i, b_{i-1}) \tag{11}$$
$$= \frac{\mathbb{P}(\vec{Y}(i)|\vec{B}', \mathcal{K}_i)}{\mathbb{P}(\vec{Y}(i)|\mathcal{K}_i, b_{i-1})} \sum_{\vec{B} \in \mathcal{B}} \mathbb{P}(\vec{B}'|\vec{B}, \mathcal{K}_i) b_{i-1}(\vec{B}),$$

where $\mathbb{P}(\vec{Y}(i)|\mathcal{K}_i, b_{i-1})$ is the normalization constant and b_{i-1} represents the belief of the agent prior to the observation $\vec{Y}(i)$, defined as a probability distribution over all possible states. Given the action $\mathcal{K}_i$, i.e. the spectrum bands sensed by the SU at time i, we employ a state estimator to identify idle bands in the spectrum. If a channel k in the estimated state vector $\hat{\vec{B}}$ is 0, the SU detects the band as being idle and accesses it for the delivery of its network flows. On the other hand, if a channel k in the estimated state vector is 1, the SU detects the band as being occupied by a PU and leaves it untouched. More details about the state estimator are discussed in the Section III. [BK: I see in your previous comment that you mentioned we use a threshold based detection to determine the occupancy after the belief update - But, in the code, I use the MAP estimator to estimate the occupancy vector based on the bands I sensed - Do you think this is incorrect?] Therefore, at time i, the probability of false alarm is defined as the ratio of the number of false alarms to the number of truly idle bands in the spectrum and the probability of missed detection is defined as the ratio of the number of missed detections to the number of truly occupied bands in the spectrum. [BK: I need advice on how to represent the definitions of P_{FA} and P_{MD} mathematically here - Do I use hypothesis based definitions or do I use indicator random variables?] Based on the number of truly idle bands detected by the SU accounting for the throughput maximization aspect of the agent's end-goal and a penalty for missed detections accounting for the incumbent non-interference constraint, the reward to the agent is modelled as

$$R(\vec{B}(i), \mathcal{K}_i) = (1 - P_{FA}(i)) + \lambda P_{MD}(i), \tag{12}$$

where $\lambda < 0$ represent the cost term penalizing the agent for missed detections, i.e. interference with the incumbent. The

action policy π of the agent maps the beliefs b_i to actions $\mathcal{K}_i$ at time i and is characterized by a Value Function

$$V^\pi(b) \;=\; \mathbb{E}_\pi\Big[\sum_{i=0}^{\infty} \gamma^i R(b_i,\; \pi(b_i)|b_0 = b)\Big], \qquad (13)$$

where $0 < \gamma < 1$ is the discount factor, $\pi(b_i)$ is the action taken by the agent at time i under policy π, and b_0 is the initial belief. The optimal policy π^* specifies the optimal action to take at the current time index assuming that the agent behaves optimally at future time indices as well. It is evident from equation (13) that we have an infinite-horizon discounted reward problem formulation and in order to solve for the optimal policy we need to solve the Bellman equation given by

$$V^*(b) = \max_{\mathcal{K}\in\mathcal{A}}\Big[\sum_{\vec{B}\in\mathcal{B}} R(\vec{B},\mathcal{K})b(\vec{B}) + \gamma\sum_{\vec{Y}\in\mathcal{Y}} \mathbb{P}(\vec{Y}|\mathcal{K},b)V^*(b_{\mathcal{K}}^{\vec{Y}})\Big].$$
$$(14)$$

Given the high dimensionality of the spectrum sensing and access problem, i.e. the number of states of the underlying MDP scales exponentially with the number of bands in the spectrum, solving equation (14) using Exact Value Iteration and Policy Iteration algorithms is computationally infeasible. Additionally, solving for the optimal policy from equation (14) requires prior knowledge about the underlying MDP's transition model. Therefore, in this paper we present a framework to estimate the transition model of the underlying MDP and then utilize this learned model to solve for the optimal policy by employing Randomized Point-Based Value Iteration techniques, namely, the PERSEUS algorithm. **[NM: All the citations should go into the file ref.bib with the specific bibtex format (you can typically get the entry from IEEE Xplore)]**

III. APPROACHES AND ALGORITHMS

A. Occupancy Behavior Transition Model Estimation

In real-world implementations of cognitive radio systems, the transition model of the occupancy behavior of the PUs is not known to the SUs in the network and needs to be learnt over time. The learnt model then needs to be fed back to the POMDP agent in order to solve for the optimal policy. The system can learn the model either before triggering or during the operation of the POMDP agent. Inherently, the approach constitutes solving a parameter estimation problem formulated as

$$A^* \;=\; \arg\max_{A}\; \mathbb{P}([\vec{Y}(i)]_{i=1}^{\tau}|A), \qquad (15)$$

which is a Maximum Likelihood Estimation (MLE) problem, where A is defined as $\mathbb{P}(\vec{B}(i+1)|\vec{B}(i))$ and τ refers to the learning period of the parameter estimator: this, as mentioned earlier, can be equal to the entire duration of the POMDP agent's interaction with the radio environment or can be a predefined parameter learning period before triggering the POMDP agent. In order to facilitate better readability, for the description of this parameter estimator, we denote $[\vec{Y}(i)]_{i=1}^{\tau}$ as $\mathbf{Y}$ and $[\vec{B}(i)]_{i=0}^{\tau}$ as $\mathbf{B}$. Re-framing (15) as an optimization of

the log-likelihood, using the definition of marginal probability, and focusing on the joint instead of the conditional, we get,

$$A = \arg\max_{A}\; \log\Big(\sum_{\mathbf{B}} \mathbb{P}(\mathbf{B},\mathbf{Y},A)\Big) \qquad (16)$$

In order to exploit the characteristics of the stated Markov model, we multiply and divide the operand of the logarithm by β which from the equality constraint of Jensen's Inequality turns out to be $\mathbb{P}(\mathbf{B}|\mathbf{Y},\hat{A})$. The optimization problem in (16) is then restated as,

$$A = \arg\max_{A}\; \sum_{\mathbf{B}} \mathbb{P}(\mathbf{B}|\mathbf{Y},\hat{A})\; \log(\mathbb{P}(\mathbf{B},\mathbf{Y},A)) \qquad (17)$$

Applying the characteristics of the Markov model discussed in Section II, we write (17) as

$$A = \arg\max_{A}\sum_{\mathbf{B}} \mathbb{P}(\mathbf{B}|\mathbf{Y},\hat{A})\sum_{L}\sum_{R}\sum_{i=1}^{\tau}\sum_{j=1}^{\tau}\mathcal{I}\log(M_R(\vec{Y}(i)))$$
$$+ \mathcal{J}\log(a_{LR}), \qquad (18)$$

where $L, R \in \{0,1\}^K$ represent iterables for the occupancy state vectors,

$$M_R(\vec{Y}(i)) = \mathbb{P}(\vec{Y}(i)|\vec{B}(i) = R), \qquad (19)$$

represents the emission model outlined in (9),

$$a_{LR} = \mathbb{P}(\vec{B}(i) = R|\vec{B}(i-1) = L, \hat{A}), \qquad (20)$$

represents the unknown transition model which is the subject of this estimation, and $\mathcal{I}$ and $\mathcal{J}$ detailed below are indicator random variables introduced to bring in specificity into the estimation procedure.

$$\mathcal{I} = \begin{cases} 1, & \text{if } \vec{Y}(i) \text{ and } \vec{B}(i) = R \\ 0, & \text{otherwise} \end{cases} \qquad (21)$$

$$\mathcal{J} = \begin{cases} 1, & \text{if } \vec{B}(i-1) = L \text{ and } \vec{B}(i) = R \\ 0, & \text{otherwise} \end{cases} \qquad (22)$$

We impose a constraint on the transition probability in (18) as

$$\sum_{R} a_{LR} = 1, \qquad (23)$$

and formulate the Lagrangian as

$$\mathcal{L} = \Big\{\sum_{\mathbf{B}} \mathbb{P}(\mathbf{B}|\mathbf{Y},\hat{A})\sum_{L}\sum_{R}\sum_{i=1}^{\tau}\sum_{j=1}^{\tau}\mathcal{I}\log(M_R(\vec{Y}(i)))$$
$$+ \mathcal{J}\log(a_{LR})\Big\} \qquad (24)$$
$$+ \sum_{L}\lambda_L(1 - \sum_{R} a_{LR}).$$

Solving for a_{LR}, we get,

$$a_{LR} = \frac{\sum_{i=1}^{\tau} \mathbb{P}(\mathbf{Y},A,\vec{B}(i) = R,\vec{B}(i-1) = L)}{\sum_{i=1}^{\tau} \mathbb{P}(\mathbf{Y},A,\vec{B}(i-1) = L)}. \qquad (25)$$

In order to further simplify (25) and bring it into an iterative algorithmic form, we introduce the forward and backward probabilities. We define the forward probability as

$$F(i, R) = \mathbb{P}([\vec{Y}(t)]_{t=1}^{i}, \vec{B}(i) = R)$$
$$= \sum_{L} \mathbb{P}(\vec{B}(i) = R, \vec{Y}(i) | \vec{B}(i-1) = L) F(i-1, L), \tag{26}$$

and the backward probability as

$$D(i, L) = \mathbb{P}([\vec{Y}(t)]_{t=i}^{\tau} | \vec{B}(i-1) = L)$$
$$= \sum_{R} \mathbb{P}(\vec{B}(i) = R, \vec{Y}(i) | \vec{B}(i-1) = L) D(i+1, R). \tag{27}$$

Using these definitions, (25) can be rewritten as,

$$a_{LR} = \frac{\sum_{i=1}^{\tau} F(i-1, L) M_R(\vec{Y}(i)) a_{LR} D(i+1, R)}{\sum_{R} \sum_{i=1}^{\tau} F(i-1, L) M_R(\vec{Y}(i)) a_{LR} D(i+1, R)}. \tag{28}$$

[BK: Could you give me an example as to how you generally write algorithms in IEEE papers? There are many ways to do it - algorithm, algorithm2e, algorithmic, and pseudo-code. Which one do you recommend and could you please give me a sample?]

B. Occupancy Behavior State Estimation

During the reward evaluation phase of the POMDP agent at time i, the observations made based on the sensing action $\mathcal{K}_i$ are employed at a state estimator to determine the occupancy state of the spectrum bands. Based on this estimated state vector, we formulate the false alarm and missed detection metrics which allow us to capture the throughput maximization and PU non-interference requirements essential for the operation of our POMDP agent. We formulate the state estimation problem as

$$\vec{B}(i)^* = \arg\max_{\vec{B}} \ \mathbb{P}(\vec{B} | \vec{Y}(i)), \tag{29}$$

which is a Maximum-A-Posteriori (MAP) estimation problem. This optimization problem can be restated in terms of the value functions as

$$V_{i,k}^r = \max_{\tilde{\mathbf{B}}_{[t-1,k-1]}} \mathbb{P}(\tilde{\mathbf{Y}}_{[t-1,k-1]}, \tilde{\mathbf{B}}_{[t-1,k-1]}, Y_k(i), B_k(i)), \tag{30}$$

where for the estimation of the occupancy in spectrum band k at time i,

$$\tilde{\mathbf{Y}}_{[t-1,k-1]} \equiv \{ [\vec{Y}_v(i)]_{v=1}^{k-1}, \ [\vec{Y}_k(t)]_{t=1}^{t-1} \} \tag{31}$$

denotes the set of all essential past observations which for readability purposes is denoted simply as $\tilde{\mathbf{Y}}$ and

$$\tilde{\mathbf{B}}_{[t-1,k-1]} \equiv \{ [\vec{B}_v(i)]_{v=1}^{k-1}, \ [\vec{B}_k(t)]_{t=1}^{t-1} \} \tag{32}$$

denotes set of all essential past states which is henceforth simply referred to as $\tilde{\mathbf{X}}$ for readability. Applying the characteristics of the Markov model detailed in (7) to (30), we get

$$V_{i,k}^r = M_r(Y_k(i)) \max_{\tilde{\mathbf{B}}} \mathbb{P}(B_k(i) = r | B_k(i-1) = m,$$
$$B_{k-1}(i) = n) \mathbb{P}(\tilde{\mathbf{Y}}, \tilde{\mathbf{B}}), \tag{33}$$

which can be simplified further to show that,

$$V_{i,k}^r = M_r(Y_k(i)) \max_{m,n} \ a_{mnr} V_{i-1,k}^m V_{i,k-1}^n, \tag{34}$$

where

$$a_{mnr} = \mathbb{P}(B_k(i) = r | B_k(i-1) = m, B_{k-1}(i) = n), \tag{35}$$

which can be evaluated from the estimated transition model. Equation (34) corresponds to the forward recursion aspect of the double Markov chain Viterbi algorithm. Next, similar to the backtracking procedure in the one dimensional (single Markov chain) Viterbi algorithm, the Trellis diagram is traversed backwards to recover the most probable previous neighbours of $B_k(i)$. This is done recursively until the entire Trellis diagram has been traversed to yield the most probable state sequence, i.e. the Viterbi path. Mathematically, the backtracking step with respect to the neighbours of $B_k(i)$ is represented as

$$m^*, n^* = \arg\max_{m,n} \ a_{mnr} V_{i-1,k}^m V_{i,k-1}^n, \tag{36}$$

where m^* is the most probable state of channel k in time index $i-1$ and n^* is the most probable state of channel $k-1$ in time index i, given that channel k in time index i is in state r; $m, n, r \in \{0, 1\}$. [BK: Add the algorithm block!]

C. The PERSEUS Algorithm

As discussed in Section II of this article, solving the Bellman equation (14) for POMDPs with large state and action space using direct value iteration and policy iteration techniques is computationally infeasible [BK: citation]. Hence, we resort to approximate value iteration techniques [BK: citation] to ensure that the system scales well to a large number of bands in the spectrum of interest. For infinite-horizon POMDPs, V^* in (14) can be approximated by a Piece-Wise Linear and Convex function (PWLC) [BK: citation]. The core idea behind the PERSEUS algorithm is that the value function in time index i can be parameterized by a set of hyperplanes $\{\vec{\alpha}_i^u\}$, $u = \{0, 1, 2, \ldots, |V_i|\}$, each of which represents a region of the belief space for which it is the maximizing element. The backup step is defined as the process of determining the optimal hyperplane out of the set of available hyperplanes in time index i as

$$\vec{\alpha}_i(b) = \arg\max_{\vec{\alpha}_i^u} b \cdot \vec{\alpha}_i^u, \tag{37}$$

which implies that,

$$V_i(b) = \max_{\vec{\alpha}_i^u} b \cdot \vec{\alpha}_i^u,$$
$$\pi_i(b) = a(\vec{\alpha}_i^{u*}), \tag{38}$$

where $a(\vec{\alpha}_i^{u*})$ refers to the action corresponding to the optimal hyperplane. The PERSEUS algorithm constitutes an Exploration phase wherein the POMDP agent randomly interacts with the radio environment to collect a set of so-called reachable beliefs which are to be improved and numerous iterative Backup stages. In each backup stage, the agent samples an unimproved belief point b uniformly at random from the set of unimproved points and performs a backup on this sampled belief point according to (37) to determine the

optimal hyperplane $\vec{\alpha}$. Considering an arbitrary time index i, if $V_{i+1}(b) = b \cdot \vec{\alpha} \geq V_i(b)$, then the belief point b is said to be improved along with any other belief points b' in the unimproved set for which $V_{i+1}(b') = b' \cdot \vec{\alpha} \geq V_i(b')$. If $V_{i+1}(b) = b \cdot \vec{\alpha} < V_i(b)$, then a copy of the maximizing hyperplane for $V_i(b)$ is used for $V_{i+1}(b)$ and the belief point b is then removed from the set of unimproved points. The backup stage continues until the set of unimproved points is empty and the agent performs a series of backup stages until there the number of policy changes between iterations is below a specified threshold η. [BK: Add the algorithm block!]

The belief update procedure outlined in (11) is an essential aspect of the PERSEUS algorithm which can turn into a performance bottleneck for large state spaces due to the inherent iteration over all possible states. In order to circumvent this problem, we fragment the spectrum into much smaller, independent sets of correlated channels and then run the PERSEUS algorithm on these fragments by leveraging multi-processing and multi-threading tools available at our disposal in software frameworks. Furthermore, we avoid iterating over all possible states and allow only those state transitions we deem to be the most probable - for example, we allow only those state transitions that involve a Hamming distance of up to 3 between the previous state vector and the current state vector in an 18 channel radio environment.