Opportunistic Spectrum Access in Wireless Communications with Network Coding

Anthony Fanous†, Yalin E. Sagduyu‡, and Anthony Ephremides†

†Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742, USA
‡Intelligent Automation Inc., Rockville, MD 20855, USA
Email: †{afanous, etony}@umd.edu, ‡ysagduyu@i-a-i.com

***Abstract*—We consider opportunistic spectrum access on a channel with stochastic traffic carried by network coding. We show how a secondary user (SU) can leverage the structure induced by block-based network coding on a primary user's (PU) channel to maximize the throughput. Network coding enhances transmission efficiency and when applied on a PU channel, it can extend spectrum availability for the SU. We study the additional gain of spectrum predictability from network coding and show that the SU can more reliably detect the idle spectrum under sensing errors if the PU channel carries network-coded transmissions even when the channel utilization remains the same. The SU maximizes the throughput by first learning the PU spectrum parameters with the Baum-Welch algorithm and then tracking the spectrum holes with the Partially Observable Markov Decision Process (POMDP) algorithm. We show that network coding renders the spectrum more predictable, which leads to a higher SU throughput.**

***Index Terms*—Opportunistic spectrum access, spectrum sensing, network coding, throughput optimization, POMDP.**

I. Introduction

Cognitive radio network paradigm aims at increasing the spectrum utilization by allowing secondary (unlicensed) users to access the licensed spectrum while guaranteeing some protection metric (e.g., throughput or interference level) to the primary (licensed) users' transmissions [1]. For opportunistic access to the licensed spectrum, the secondary users (SUs) need to capture the idle times in primary user (PU) transmissions (also called "spectrum holes") with reliable spectrum sensing. Although several sensing techniques have been proposed in the literature (e.g., energy detectors and cyclostationary detectors [2]), spectrum sensing remains challenging due to channel impairments such as fading, path loss, and shadowing, which may result in erroneous decisions Therefore, it is important to mitigate sensing errors for both PU protection and to achieve high SU throughput performance.

Spectrum sensing is typically applied in a "memoryless" way without exploiting the possible correlation between the PU spectrum states. Spectrum sensing techniques that exploit the memory of PU spectrum dynamics have been proposed [3], [4], but they are usually limited to the transition of the PU spectrum between two (idle/busy) states. In this paper, we study the additional gains of (i) spectrum availability and (ii) spectrum predictability resulting from the correlation of the PU states induced by network coding. In a cognitive radio system, the spectrum availability to the SUs can increase if the PU uses network coding for their transmissions. Network coding has emerged as a powerful scheme to improve the throughput by coding over packet traffic [5] and it is known that it can improve the throughput in both multi-hop [6] and single-hop [7] networks with low complexity implementation. Therefore, it is natural to consider its application to PU communications and the side effect will be more idle time available to the SUs (i.e., higher spectrum availability). The second effect of network coding is that it "shapes" the spectrum and induces a structure to the PU channel occupancy states such that network-coded transmissions occur in batches rather than sporadically (i.e., higher spectrum predictability). With network coding, the busy period on the PU channel is lower-bounded by the coding block size, K, and the idle period must accumulate a block of K packets to initiate transmissions.

In this paper, we study how the SU can exploit the structure of the PU's idle and busy periods induced by network coding to reliably learn the PU spectrum characteristics and effectively mitigate spectrum sensing errors. Since network coding results in blocks of busy and idle slots rather than scattered over time, as illustrated in Figure 1, the PU spectrum becomes more predictable with network coding. Based on the structural properties of network-coded spectrum we study constructive algorithms for the SU that can first estimate the PU spectrum parameters in single-hop communications and then identify the idle slots on PU spectrum with high accuracy. Note that the spectrum predictability gain (i.e., "shaping effect") is present even when there is no spectrum availability gain, e.g., when the PU transmitter has perfect channels to multiple receivers or there is a single receiver with an imperfect channel where network coding has no throughput advantage over retransmissions. The goal of this paper is to exploit these two spectrum effects to mitigate spectrum sensing errors for improving the SU throughput

Network coding has been studied for spectrum sensing

This material is based upon work supported by the Air Force Office of Scientific Research under STTR Contracts FA9550-11-C-0006 and FA9550-12-C-0037. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Air Force Office of Scientific Research. This work was supported in part by MURI grant W911NF-08-1-0238, NSF grant CCF0905204, NSF grant CNS1147730, and ONR grant N000141110127.

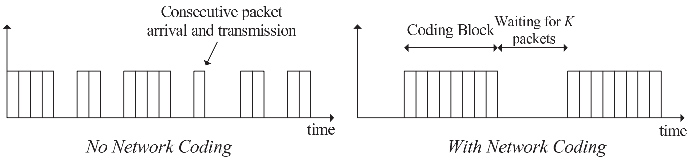

Fig. 1: Busy/idle periods with and without network coding.

purposes mainly in two directions. First, in [8], network coding was used in conjunction with collaborative spectrum sensing to help efficiently disseminate control information among the SUs. Second, in [9], the correlation among PU spectrum states due to network coding was used by the SUs to track multiple PU channels (by assuming that a busy slot will be more likely followed by another busy slot) and to quickly identify an idle channel. However, the model in [9] assumes that the SU has perfect sensing capability and can correctly distinguish an idle slot from a busy one on the channel it chooses to sense.

In this paper, we relax the idealistic assumption of perfect sensing by studying the practical case with possible spectrum sensing errors. While providing some degree of protection to the PU, the SU pursues the objective of average throughput maximization. We build a systematic model that represents the structural effects of block-based network coding on PU spectrum. We first consider a perfect PU channel such that there is no spectrum availability gain (i.e., the fraction of the idle slot is the same independent of coding block size K), and then extend the model to imperfect PU channels (where the fraction of idle slots depends on K).

We show that a higher value of K leads to better learning of the PU spectrum structure via the Baum-Welch algorithm and then by using the POMDP algorithm we show that the spectrum predictability due to network coding applied at the PU can actually improve the SU throughput (and the gain increases with K), even when the spectrum utilization remains the same. Then, we consider imperfect channels and evaluate the throughput benefit of network coding as a combined effect of spectrum availability and predictability gains. Our results show that the benefit of using network coding by the PU is not limited to PU throughput gain, but also, if properly exploited, improves the spectrum sensing accuracy (under sensing errors) and increases the throughput for the SU by mitigating possible sensing errors. This way, the overall spectrum efficiency of a cognitive radio network can be significantly improved.

The rest of the paper is organized as follows. In Section II, we introduce the system model for spectrum sensing on a PU channel. Section III studies the benefit of spectrum predictability gain for the SU throughput maximization. We evaluate the additional gain of spectrum availability for the SU throughput in Section IV. Section V concludes the paper.

II. System Model

We consider a cognitive radio network consisting of one PU and one SU. The SU tries to detect the idle periods on the PU's channel for opportunistic access. The PU channel state (busy/idle) is random due to random packet arrivals and possible (random) transmission failures. Time is slotted with slot duration equal to one packet transmission. The SU senses the spectrum (e.g., with energy detector) at the beginning of every slot to detect whether the PU channel is idle or not. Based on channel sensing results (subject to sensing errors) and possible prior knowledge of PU channel statistics, the SU decides on whether to transmit or not. We consider the problem of the SU throughput maximization while guaranteeing some level of throughput to the PU's transmissions.

The PU generates (or receives) packets according to a stationary process and buffers them until K packets are accumulated in its queue. The PU then codes the block of K packets linearly and transmits K coded packets. The PU and its receiver(s) agree on a set of linearly independent coding coefficients such that K successful transmissions are needed at a receiver to decode a block of K packets (or random network coding with sufficiently large field size is considered). The state of the PU channel (idle/busy) is assumed to be fixed over a slot duration and varies between slots according to a Markov chain, which models the correlation between the PU states. Exact modeling of the queue dynamics with network coding leads to an infinite-state Markov chain, which renders it hard to be learned or tracked at the SU, especially over short time intervals. To avoid the complicated queue evolution, we use several approximate models for the idle/busy periods of a PU with network-coded transmissions.

For the PU systems without network coding, the busy/idle periods have been observed via channel measurements [10] to follow a two-state Markov chain (shown in Figure 2) and this model has been widely used in spectrum sensing (e.g., [3], [4]). Our model can be viewed as a natural extension of this two-state model to network coding. We start with the case of a perfect PU channel that requires K transmissions (in K slots) to deliver K network-coded packets. In Section IV, we will extend the model to random packet erasures on the PU channel where the PU continues transmitting coded packets (possibly each of them multiple times) until K transmissions are successfully received.

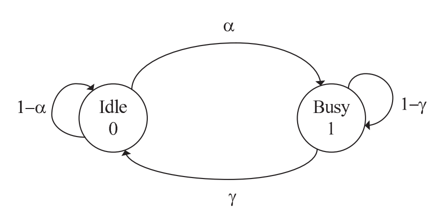

Fig. 2: PU spectrum dynamics without network coding.

For the case of a perfect PU channel, the idle/busy states of the PU are modeled by the Markov chain shown in Figure 3. States 0 to m ($0 \leq m \leq K-1$) correspond to idle states (i.e., buffering K packets while waiting for transmission) and states $m+1$ to $m+K$ correspond to busy states (i.e., transmitting K coded packets). The transition probabilities $p_{i,j}$ for $0 \leq i \leq j \leq m$ depend on the arrival process at the PU's buffer. After receiving K packets, the busy period has a duration of K slots under error-free channels.

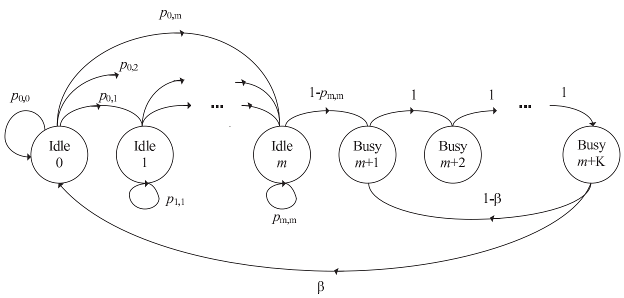

Fig. 3: PU spectrum dynamics with network coding.

We note two special structures of the idle periods in the PU spectrum. One extreme corresponds to $m = K - 1$ with Bernoulli arrivals (corresponding to $p_{i,j} = 0$ for $j > i + 1$) and models the case where packets arrive at the PU queue one by one according to a Bernoulli process with average rate λ such that the PU needs to wait for at least K slots until K packets are in its buffer (this case is a good approximation for low arrival rates), and thus the duration of the idle period is at least K slots. The other extreme corresponds to $m = 0$ and models the case when packets arrive at the PU in batches, each with K packets. Batches arrive one at a time according to a Bernoulli process with rate λ and hence, $p_{0,0} = 1 - \lambda$. This models the case when the PU is an intermediate relay node in a network and receives packets from K neighboring nodes (this case is a good approximation for high arrival rates). Compared to $m = K - 1$, the PU does not have to wait for at least K slots to start network-coded transmissions if $m = 0$. In this paper, we consider the case with $m = 0$ that has a rather less obvious structure of the idle periods in the PU spectrum and show that even in this case the spectrum predictability allows the SU to effectively mitigate channel sensing errors.

From the structure of the Markov chain, we expect that with larger K, the SU can more reliably detect the idle PU slots through better tracking of the PU states, since it is more likely that a busy slot is followed by another busy slot due to block transmission structure of coded packets. For instance, if the SU can correctly detect the first slot of a busy period, it knows (without ambiguity) that the following $K - 1$ slots will also be busy. On the other hand, if $K = 1$, the busy/idle sequence, as modeled in Figure 2, involves less structure and in particular, for $\alpha = \gamma = 0.5$, it forms an i.i.d. sequence with no memory between the PU states that the SU could use to track and predict the sequence.

The idle state of the Markov chain in Figure 3 with $m = 0$ has stationary probability $\pi_0 = \frac{\beta}{\beta+K\lambda}$ and the channel utilization is $u = 1-\pi_0 = \frac{K\lambda}{\beta+K\lambda}$. To observe only the shaping effect of network coding in terms of spectrum predictability (by separating the effect of spectrum availability gain, which we will study later in Section IV), the parameters of the Markov chain are chosen to yield the same channel utilization for all K. This means that the fraction of the idle and busy slots is the same for all K, but the durations of the busy periods are multiples of K slots and consequently the idle periods are, on average, K times longer due to the buffering and the batch processing required by network coding. The transition probability β for an idle state to follow a busy period should decrease with λ, and we set $\beta = 1 - \lambda$ to approximate the queue behavior under network coding[1]. The Markov chain then reduces to the one shown in Figure 4, which we will use in Section III. By setting $\lambda = \frac{u}{u+K(1-u)}$, we obtain the same channel utilization u for any K and hence we only observe the shaping effect of network coding.

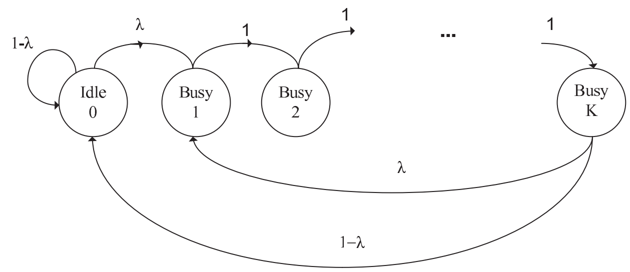

Fig. 4: PU spectrum dynamics with network coding for perfect PU channel with $m = 0$.

We assume that the spectrum sensor used at the SU has misdetection probability p_M and false alarm probability p_F. We will show that the SU can mitigate channel sensing errors by tracking the PU spectrum dynamics in contrast to memoryless sensing strategies that cannot exploit the possible correlation of the PU states over time.

III. SU Throughput Maximization on Idle Slots of the PU Channel

The objective of the SU is to maximize the average throughput that is measured as the average rate of detecting an idle slot on the PU channel. We first consider the learning phase, where the SU passively observes the channel (without transmitting) in order to infer the parameters of the PU Markov chain; then we focus on the tracking phase, where the SU actively tracks the channel for opportunistic access. We consider the Baum-Welch algorithm [11] for the learning phase and the POMDP algorithm for the tracking phase. In this section, we only study the spectrum predictability gain of network coding while we address the additional spectrum availability gain in Section IV.

A. Learning Phase

The SU learns the Markov chain parameters given the coding block size K used by the PU. We define N as the number of slots over which the SU observes the PU chain for learning and $\mathcal{S} = \{0, 1, ..., K\}$ as the state space of the PU Markov chain. The true sequence of states $s_1^N = \{s_t \in \mathcal{S} | t = 1, 2, ..., N\}$ is hidden to the SU but a sequence of corresponding sensing outcomes $y_1^N = \{y_t \in \mathcal{Y} | t = 1, 2, ..., N\}$ is available to the SU, where $\mathcal{Y}$ ={"Idle", "Busy"}.

Given only the observation sequence, the Baum-Welch algorithm generates a sequence of parameter estimates of non-decreasing likelihood values for the Hidden Markov Process

[1] In particular, this is exact behavior if the buffer size at the PU is of size K and hence batches arriving during a busy period are dropped.

(HMP). Define $\hat{\eta}_r = (\hat{\pi}_r, \hat{A}_r)$ as the estimate of the parameters of the hidden Markov chain at the rth iteration of the algorithm, where $\hat{\pi}_r$ is the estimated initial distribution of the chain and $\hat{A}_r = [\hat{a}_{ij}]_r$ is the estimated state transition matrix. The algorithm starts with an initial guess $\hat{\eta}_0 = (\hat{\pi}_0, \hat{A}_0)$ and then updates the parameter estimates by maximizing the likelihood given the observation sequence $\{y_1^N\}$. The rth iteration starts with an estimate $\hat{\eta}_{r-1}$ and estimates a new parameter set $\hat{\eta}_r$ according to

$$\hat{\eta}_r = \arg\max_{\bar{\eta}} \sum_{s_1^N} P_{\hat{\eta}_{r-1}}\left(s_1^N | y_1^N\right) \ln\left[P_{\bar{\eta}}\left(s_1^N, y_1^N\right)\right], \quad (1)$$

where $P_{\hat{\eta}_{r-1}}\left(s_1^N | y_1^N\right)$ is the probability of the state sequence s_1^N given the observation sequence y_1^N under model estimate $\hat{\eta}_{r-1}$ and $\bar{\eta}$ is the set of the feasible parameters at the rth iteration. The algorithm terminates when a convergence criterion is satisfied, e.g., when $\ln P_{\hat{\eta}_r}(y_1^N) - \ln P_{\hat{\eta}_{r-1}}(y_1^N) < \delta$ for a given threshold δ. Note that although Baum-Welch algorithm is guaranteed to converge, it might converge to a local optimum and therefore different initial guesses may be needed.

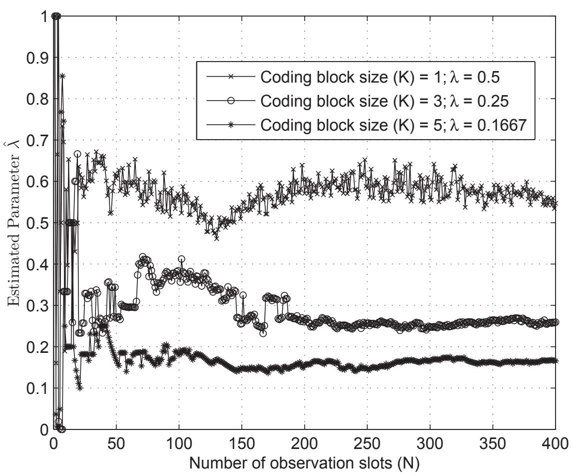

Fig. 5: Estimated parameter $\hat{\lambda}$ vs. N.

We give an example in Fig. 5 for estimating the batch arrival rate λ with threshold $\delta = 10^{-4}$. We consider a spectrum sensor with $p_M = 0.2$ and $p_F = 0.2618$ (these values are obtained from energy detector with $n = 10$ samples/slot under additive Gaussian model with 0 dB signal-to-noise ratio; however, our approach works for any spectrum sensor with the same p_M and p_F values). The channel utilization is fixed to $u = 0.5$ by properly selecting the value of λ as described in Sec II. Figure 5 shows that a higher value of K leads to a better estimate of λ for the same number of observations, N. When $K = 1$ and $K = 3$, for $N = 400$, the estimate $\hat{\lambda}$ does not exactly match the true values of λ, which are 0.5 and 0.25, respectively, while it matches the true value of λ when $K = 5$. This confirms that the PU spectrum structure due to network coding improves the estimate of the PU Markov chain parameters, which we will use next for better tracking of the PU spectrum.

B. Tracking Phase

Next, we focus on the tracking phase, where we assume that the SU perfectly knows the parameters of the PU Markov chain. At the beginning of each slot, the SU senses the channel and then chooses between two actions: either to transmit or to remain silent. With rewards incurred for different SU actions and different PU states, the optimal policy can be found through POMDP formulation. The SU maintains a belief vector about the state of the Markov chain for the PU spectrum. Each component of the belief vector represents the conditional probability that the PU Markov chain is in a certain state given the decision and observation history. The belief vector is updated based on the sensing outcome and the feedback (ACK/NACK) received upon transmission. The POMDP formulation fully captures the interplay between sensing and transmission decisions.

The problem of maximizing the SU throughput subject to some PU protection constraint (for instance, a target misdetection probability at the PU receiver) leads to a constrained POMDP formulation. For constrained POMDPs, randomized policies may be needed for optimality, while an optimal deterministic policy always exists for unconstrained POMDPs. Instead of solving the complicated constrained POMDP problem, we use an unconstrained POMDP formulation where the PU is protected by using a reward function at the SU that is the weighted sum of the PU and SU throughputs. By adjusting the weight, the PU is supported with different throughput values. The details are presented in the reward formulation.

As studied in [3], [4], we assume for simplicity that the PU's traffic dynamics are independent of the actions taken at the SU (i.e., PU unsuccessful packets are not retransmitted and hence do not affect the Markov chain evolution) and we consider the extended Markov chain structure in Figure 4 to represent network coding effects. We denote by $X_t \in \{0, 1, ..., K\}$ the state of the PU in time slot t.

1) Actions: Two actions are possible at the SU in each slot t: to remain silent ($A_t = 0$) or to transmit ($A_t = 1$).

2) Rewards: When the PU spectrum is at state X_t and action A_t is taken by the SU, the reward is given by

$$R(X_t, A_t) = \begin{cases} 0, & \text{if } X_t = 0,\ A_t = 0 \\ w\,\tilde{r}_P, & \text{if } X_t \neq 0,\ A_t = 0 \\ (1-w)\tilde{r}_S, & \text{if } X_t = 0,\ A_t = 1 \\ 0, & \text{if } X_t \neq 0,\ A_t = 1 \end{cases}, \quad (2)$$

where $\tilde{r}_P$ and $\tilde{r}_S$ are the rates achieved by the PU and SU, respectively, when they do not interfere. Simply, we assume $\tilde{r}_P = \tilde{r}_S = 1$ (coded) packet/slot. The weight w represents the relative importance of the PU throughput and is used for the PU protection. Choosing $w = 1$ gives full priority to the PU throughput (i.e., full PU protection), while $w = 0$ favors the SU throughput (i.e., no PU protection). By varying w between 0 and 1, we can reach different degrees of PU protection corresponding to different PU throughputs. Note that we are assuming a collision channel, that is, when both the PU and the SU transmit simultaneously, a collision occurs and both packets are lost. However, a similar approach can be taken for the case with multi-packet reception capability. If the PU codes over K packets, any coded packet loss due to collision may prevent the PU receiver from decoding the entire block of coded packets (since PU's failed packets are not retransmitted). This can be overcome if the PU codes over $K-1$ packets, transmits this block, and then at the Kth transmission, either retransmits one of these coded packets if a collision occurred, or else transmits the remaining uncoded packet. As we will see later, the optimal access policy at the SU guarantees $K-1$ collision-free transmissions out of K PU transmissions during a busy period, and hence, at least $K-1$ PU coded packets can be successfully delivered during every PU busy period. This validates Eq. (2), where the PU has one packet delivered when it is busy and no collision occurs.

3) Spectrum Sensing: Although some level of PU protection can be achieved solely by adjusting w in the reward function, spectrum sensing leads to better inference of the PU state and consequently higher SU throughput for the same level of PU protection (same PU throughput). Assume that the spectrum sensing scheme has a misdetection probability p_M, which corresponds to some false alarm probability p_F.

4) Channel Feedback from SU receiver: If the SU chooses to transmit, an error-free feedback message is sent from the SU receiver to the SU transmitter indicating whether the packet was successfully received or not. This feedback message also reveals about the idle/busy state of the PU spectrum since under the collision channel model considered, if an ACK (or NACK) is received at the end of a slot, the SU learns that the PU was idle (or busy) during that slot. This feedback is then used by the SU for updating the belief in the next slot.

5) Observations and Belief Vector: Since the true state of the PU spectrum cannot be exactly observed from channel sensing results because of possible sensing errors, the SU maintains a belief about the state of the PU spectrum. For POMDP problems, the belief is a sufficient statistic for deciding on the action given all the past observations and actions [12]. Given the spectrum sensing observation and the transmission feedback, the SU updates its belief regarding the state of PU spectrum. We denote by Λ_t the $(K+1)\times 1$ belief vector of the PU state in time slot t, where the mth component $\Lambda_t(m)$ denotes the belief in time slot t that the Markov chain in Figure 3 is in state m, where $0 \leq m \leq K$. Note that the first component of the belief vector is $\Lambda_t(0)$.

(a) Under the action $A_t = 0$: The SU chooses not to transmit. No channel feedback is observed and the belief in the following slot is updated solely based on the channel sensing outcome in that slot. The observation is either "Busy" or "Idle". Denote $1-x$ by $\overline{x}$. Given a belief vector Λ_t, the probabilities of observing the outcome "Busy" and "Idle" are given by Eqs. (3) and (4) respectively.

$$\Pr[\text{Busy}|\Lambda_t] = \Lambda_t(0)\left[\lambda\,\overline{p_M} + \overline{\lambda}\,p_F\right] + \Lambda_t(K)\left[\lambda\,\overline{p_M} + \overline{\lambda}\,p_F\right] + \overline{p_M}\sum_{m=1}^{K-1}\Lambda_t(m), \tag{3}$$

$$\Pr[\text{Idle}|\Lambda_t] = \Lambda_t(0)\left[\lambda\,p_M + \overline{\lambda}\,\overline{p_F}\right] + \Lambda_t(K)\left[\lambda\,p_M + \overline{\lambda}\,\overline{p_F}\right] + p_M\sum_{m=1}^{K-1}\Lambda_t(m), \tag{4}$$

The belief updates under action $A_t = 0$, given observations $O(A_t) = \text{Idle}$ and $O(A_t) = \text{Busy}$ are given by Eqs. (5) and (6), respectively.

$$\Lambda_{t+1}(m) = \Pr[X_{t+1} = m|A_t = 0, \Lambda_t, O(A_t) = \text{Idle}] = \frac{\left[\overline{p_F}\,\mathbf{1}[m=0] + p_M\mathbf{1}[m\neq 0]\right]\Gamma_m}{\sum_{m=0}^{K}\left[\overline{p_F}\,\mathbf{1}[m=0] + p_M\mathbf{1}[m\neq 0]\right]\Gamma_m}, \tag{5}$$

$$\Lambda_{t+1}(m) = \Pr[X_{t+1} = m|A_t = 0, \Lambda_t, O(A_t) = \text{Busy}] = \frac{\left[p_F\mathbf{1}[m=0] + \overline{p_M}\,\mathbf{1}[m\neq 0]\right]\Gamma_m}{\sum_{m=0}^{K}\left[p_F\mathbf{1}[m=0] + \overline{p_M}\,\mathbf{1}[m\neq 0]\right]\Gamma_m}, \tag{6}$$

where $\mathbf{1}[]$ is indicator function and Γ_m is given by

$$\Gamma_m = \sum_{i=1}^{K-1}\Lambda_t(i)\mathbf{1}[m=i+1] + \mathbf{1}[m=1]\left[\Lambda_t(0)\lambda + \Lambda_t(K)\lambda\right] + \mathbf{1}[m=0][\Lambda_t(0)\overline{\lambda} + \Lambda_t(K)\overline{\lambda}]. \tag{7}$$

(b) Under the action $A_t = 1$: The SU chooses to transmit. An (ACK/NACK) feedback is sent from the SU receiver to the SU transmitter over a dedicated control channel at the end of the slot. The possible observations in this case are (ACK, Busy), (ACK, Idle), (NACK, Busy) and (NACK, Idle). The (ACK/NACK) feedback is observed at the end of slot t, while the "Busy" or "Idle" outcome is observed after sensing in slot $t+1$. The probabilities of these observations under action $(A_t = 1)$ are given by Eqs. (8)-(11). The belief under different observations is updated as given by Eqs. (12)-(16).

$$\Pr[(\text{ACK}, \text{Busy})|\Lambda_t] = \Lambda_t(0)\left[\lambda\,\overline{p_M} + \overline{\lambda}\,p_F\right], \tag{8}$$

$$\Pr[(\text{NACK}, \text{Busy})|\Lambda_t] = \overline{p_M}\sum_{m=1}^{K-1}\Lambda_t(m) + \Lambda_t(K)\left[\lambda\,\overline{p_M} + \overline{\lambda}\,p_F\right], \tag{9}$$

$$\Pr[(\text{ACK}, \text{Idle})|\Lambda_t] = \Lambda_t(0)\left[\lambda p_M + \overline{\lambda}\,\overline{p_F}\right], \tag{10}$$

$$\Pr[(\text{NACK}, \text{Idle})|\Lambda_t] = p_M\sum_{m=1}^{K-1}\Lambda_t(m) + \Lambda_t(K)\left[\lambda p_M + \overline{\lambda}\,\overline{p_F}\right]. \tag{11}$$

(*i*) If $O(A_t) =$ (ACK, Busy),

$$\Lambda_{t+1}(m) = \frac{\overline{p_M}\,\mathbf{1}[m=1]\left[\lambda\,\mathbf{1}[m=1] + \overline{\lambda}\,\mathbf{1}[m=0]\right]}{p_F\overline{\lambda} + \lambda\,\overline{p_M}} + \frac{p_F\mathbf{1}[m=0]\left[\lambda\mathbf{1}[m=1] + \overline{\lambda}\,\mathbf{1}[m=0]\right]}{p_F\,\overline{\lambda} + \lambda\,\overline{p_M}}. \tag{12}$$

(*ii*) If $O(A_t) =$ (NACK, Busy),

$$\Lambda_{t+1}(m) = \frac{\left[\overline{p_M}\mathbf{1}[m\neq 0] + p_F\mathbf{1}[m=0]\right]\Psi_m}{\sum_{m=0}^{K}\left[p_F\mathbf{1}[m=0] + \overline{p_M}\mathbf{1}[m\neq 0]\right]\Psi_m}, \tag{13}$$

where

$$\Psi_m = \sum_{i=1}^{K-1} \Lambda_t(i)\mathbf{1}[m=i+1] + \mathbf{1}[m=0]\Lambda_t(K)\overline{\lambda} + \mathbf{1}[m=1]\Lambda_t(K)\lambda. \quad (14)$$

(*iii*) If $O(A_t) =$ (ACK, Idle),

$$\Lambda_{t+1}(m) = \frac{p_M \mathbf{1}[m=1]\left[\lambda\mathbf{1}[m=1]+\overline{\lambda}\,\mathbf{1}[m=0]\right]}{\overline{p_F}\,\overline{\lambda}+\lambda p_M} + \frac{\overline{p_F}\,\mathbf{1}[m=0]\left[\lambda\mathbf{1}[m=1]+\overline{\lambda}\,\mathbf{1}[m=0]\right]}{\overline{p_F}\,\overline{\lambda}+\lambda p_M}. \quad (15)$$

(*iv*) If $O(A_t) =$ (NACK, Idle),

$$\Lambda_{t+1}(m) = \frac{\left[p_M\mathbf{1}[m\neq 0]+\overline{p_F}\,\mathbf{1}[m=0]\right]\Psi_m}{\sum_{m=0}^{K}\left[\overline{p_F}\,\mathbf{1}[m=0]+p_M\mathbf{1}[m\neq 0]\right]\Psi_m}. \quad (16)$$

6) Policy: The SU policy is a mapping from the belief space to the action space. The optimal policy maximizes the expected discounted reward and is given by

$$\pi^* = \arg\max_{\pi} \mathbb{E}_\pi\left[\sum_{t=1}^{\infty}\xi^t R\left(X_t, A_t\right)|\Lambda_1\right], \quad (17)$$

where ξ is the discount factor ($\xi < 1$), which describes the importance of the future reward relative to the immediate reward, and Λ_1 is the initial belief vector, which is equal to the stationary distribution of the chain.

Let $V(\Lambda_t)$ denote the value function, which is defined as the maximum expected reward that can be incurred starting from time slot t given belief Λ_t. The value function must satisfy for all t the Bellman equation in dynamic programming given by

$$V(\Lambda_t) = \max_{A_t\in\{0,1\}}\left[R_{A_t}(\Lambda_t) + \xi\sum_{O(A_t)}\Theta\left(O(A_t), A_t, \Lambda_t\right)\right],$$

where $\Theta(O(A_t), A_t, \Lambda_t) = \Pr[O(A_t)|\Lambda_t]V(\Phi(\Lambda_t|A_t, O(A_t)))$, $R_{A_t}(\Lambda_t)$ is the expected immediate reward in slot t under action A_t, $O(A_t)$ represents the observation under action A_t and the function $\Phi(\Lambda_t|A_t, O(A_t))$ represents the belief update under action A_t and observations $O(A_t)$. For action $A_t \in \{0,1\}$, $R_{A_t}(\Lambda_t)$ is given by $R_0(\Lambda_t) = w\,\tilde{r}_P(1-\Lambda_t(0))$ and $R_1(\Lambda_t) = (1-w)\tilde{r}_S\Lambda_t(0)$, where $\Lambda_t(0)$ is the first component of the belief vector Λ_t, which represents the belief that the PU state is idle at time t.

POMDPs are PSPACE-hard problems [13]. Although several heuristics have been proposed to compute suboptimal policies for large dimensional POMDPs, we limit ourselves to small values of K and compute the optimal policy using the value iteration algorithm applied to a discrete finite uniform grid in the belief space. Once the optimal SU transmission policy is computed, we run a system simulation to compute the PU and SU throughputs.

In Figure 6, we vary the PU protection weight parameter w and show the relationship of the SU throughput, r_S, and PU throughput, r_P, jointly achieved under the POMDP algorithm for fixed channel utilization $u = 0.5$ and discount factor $\xi = 0.9$. For the same value of r_P, r_S increases with coding block size K. For r_S to reach its maximum value $1-u = 0.5$, the value of r_P must drop to zero when $K = 1$, while for $K = 2$ or $K = 3$ the PU can still sustain a non-zero throughput r_P while $r_S = 0.5$. This can be explained as follows. For $u = 0.5$ and $K = 1$ ($\lambda = 0.5$), the busy/idle states of the PU spectrum form an i.i.d. sequence and hence the spectrum cannot be predicted at the SU. For this case, three possible transmission strategies at the SU are as follows. The first strategy is to transmit in all slots irrespective of the PU and this corresponds to $r_S = 1-u = 0.5$ and $r_P = 0$ due to continuous collisions. The second strategy is to trust its sensing outcome in each slot. This corresponds to $r_S = (1-u)(1-p_F)$ and $r_P = u(1-p_M)$ and is given by the breaking point on the curve of $K = 1$. The third strategy is to remain silent at all slots and corresponds to $r_P = u = 0.5$ and $r_S = 0$. By time sharing between the first and second strategies or between

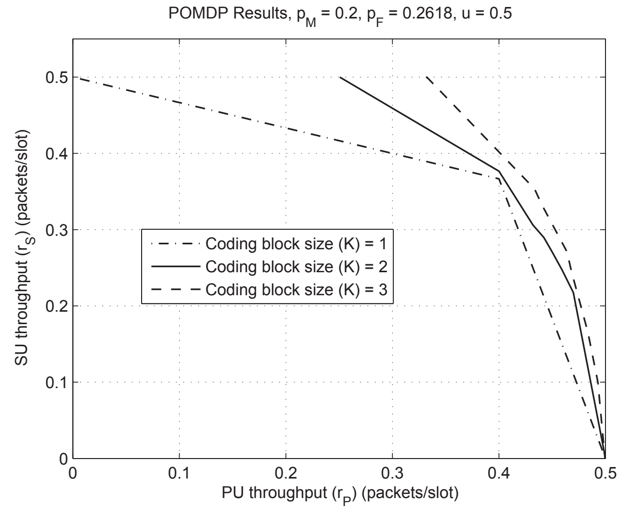

Fig. 6: Throughput performance of the POMDP algorithm.

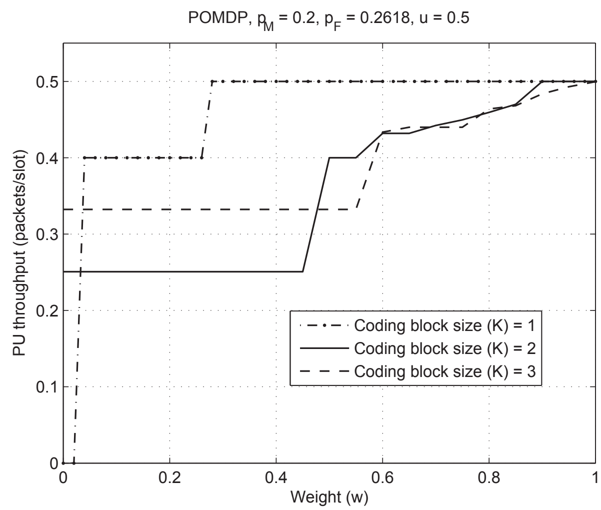

Fig. 7: PU throughput as a function of the protection weight w for the POMDP algorithm.

the second and third strategies, the points on the linear parts on the curve of $K=1$ in Figure 6 can be achieved. On the other hand, increasing K to 2 or 3 introduces more correlation (hence more memory) to the PU states and hence the Markov chain becomes more amenable to be tracked at the SU facing the sensing errors.

For the SU to achieve the maximum possible throughput $r_S = 1-u$, it can keep transmitting until a collision occurs. Then, it remains silent for the following $K-1$ slots as the PU channel will be busy in these slots and hence both the PU and SU packets would be lost in collisions if the SU transmits. This corresponds to PU throughput $r_P = \left(\frac{K-1}{K}\right) u$. This strategy converges to the optimal throughput pair ($r_P = u, r_S = 1-u$) as $K \to \infty$. For $r_S < 1-u$, this strategy is used at the SU while also remaining silent over more slots based on the tracking outcome. This provides the PU with more collision-free slots leading to a higher value of r_P and justifies our earlier claim that the optimal policy guarantees at least $K-1$ collision free PU transmissions during each PU busy period.

Figure 7 shows how to choose the weight w defined in the reward function of the POMDP formulation to provide some level of protection to the PU. For a given target PU throughput, the corresponding weight w can be found from Figure 7 and this weight is used in computing the optimal access policy at the SU. Note that the horizontal lines in Figure 7 correspond to one point in Figure 6, i.e., there may exist several weight values w leading to the same optimal policy and consequently to the same pair of PU and SU throughputs, r_P and r_S.

IV. Joint Effect of Spectrum Availability and Spectrum Predictability

So far we assumed a perfect PU channel such that the channel utilization remains the same (independent of K) and the only effect that we considered was the spectrum predictability gain. Next, we add the spectrum availability gain that network coding at the PU can provide for the case of an imperfect multicasting channel. Again, the PU needs K successful transmissions to deliver K packets but it may need more than K transmissions due to packet erasures.

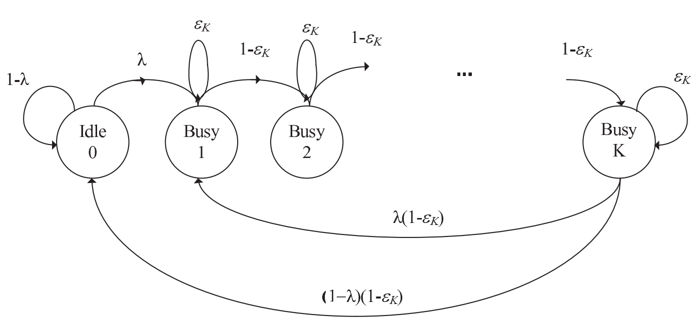

Fig. 8: PU spectrum dynamics with network coding under the erasure model.

The new Markov chain for the PU spectrum dynamics that approximately models the effects of packet erasures is shown in Figure 8. There is a transition from state i to $i+1$ (where $i = 1, 2, ..., K-1$) with probability $1-\epsilon_K$; otherwise, the state remains the same with probability ϵ_K. Note that the effective erasure probability ϵ_K depends on the PU coding block size K, the number of PU receivers (in the multicasting setting), and the success probability over each of the PU channels. The stationary distribution of the Markov chain is given by $\pi_0 = \frac{(1-\lambda)(1-\epsilon_K)}{K\lambda+(1-\lambda)(1-\epsilon_K)}$ and $\pi_m = \frac{\lambda\pi_0}{(1-\lambda)(1-\epsilon_K)}$ for $1 \leq m \leq K$. The channel utilization is $u(K) = \frac{K\lambda}{K\lambda+(1-\lambda)(1-\epsilon_K)}$.

We fix the average arrival rate in packets/slot to α for all K by choosing $\lambda = \alpha/K$ batches/slot, where the batch size is K. In a general multicasting system with erasure channels, the PU transmitter using network coding achieves a higher throughput for the same channel utilization as K increases, which is equivalent to achieving the same throughput with smaller channel utilization as K increases. This way, the PU achieves more successful packet deliveries over shorter busy periods. Hence, by fixing the maximum PU throughput for all K, we achieve a higher spectrum availability to the SU by increasing K. The same maximum possible PU throughput for all K can be achieved by setting $u(1)(1-\epsilon_1) = u(K)(1-\epsilon_K)$ leading to $\epsilon_K = \frac{K\epsilon_1-(K-1)(1-\epsilon_1)}{K-(K-1)(1-\epsilon_1)}$, where ϵ_1 is the effective erasure probability when $K = 1$. With this choice, the channel utilization $u(K) = \frac{\alpha}{\alpha+(1-\alpha/K)(1-\epsilon_K)}$ decreases as K increases, pointing to the spectrum availability gain of network coding, while the maximum PU throughput $u(K)(1-\epsilon_K)$ is the same for all K. Also, $\epsilon_K = \frac{K\epsilon_1-(K-1)(1-\epsilon_1)}{K-(K-1)(1-\epsilon_1)}$ decreases as K increases and this models more successful packet deliveries achieved by the PU for the same number of transmissions. Note that this model is only valid if $\epsilon_1 > \frac{K-1}{2K-1}$ to ensure that $\epsilon_K \geq 0$.

Next, we apply the POMDP algorithm and combine the spectrum predictability and availability gains due to network coding. The only change in the POMDP formulation is in the expected immediate reward under action $A_t \in \{0,1\}$, which is now given by $R_0(\Lambda_t) = w\, \tilde{r}_P(1-\epsilon_K)(1-\Lambda_t(0))$ and $R_1(\Lambda_t) = (1-w)\tilde{r}_S\Lambda_t(0)$, and the reward is given by

$$R(X_t, A_t) = \begin{cases} 0, & \text{if } X_t = 0,\, A_t = 0 \\ w\, \tilde{r}_P(1-\epsilon_K), & \text{if } X_t \neq 0,\, A_t = 0 \\ (1-w)\tilde{r}_S, & \text{if } X_t = 0,\, A_t = 1 \\ 0, & \text{if } X_t \neq 0,\, A_t = 1 \end{cases}. \quad (18)$$

The probability of observations and the belief updates in the presence of erasures can be derived as in the case without erasures and is omitted here for brevity. Due to the PU throughput gain of network coding, we expect that by increasing K, we get higher SU throughput (due to both spectrum predictability and availability gains). However, for the same K and larger ϵ_K, the PU remains busy for longer durations because of packet erasures and hence the SU throughput degrades.

In the numerical results, we use $\alpha = 0.5$, $p_M = 0.2$, $p_F = 0.2618$ and $\tilde{r}_P = \tilde{r}_S = 1$. The value of ϵ_1 is chosen first and then the values of ϵ_K, $K \geq 2$ are determined accordingly to yield the same maximum PU throughput. For $\epsilon_1 = 0.5$, we obtain $\epsilon_2 = 0.333$ and $\epsilon_3 = 0.25$, while for $\epsilon_1 = 0.6$ we obtain $\epsilon_2 = 0.5$ and $\epsilon_3 = 0.4545$. Figure 9 shows the effect of erasures on learning the HMP parameters. A larger value of erasure probability ϵ_K makes the PU Markov chain harder to learn because of the larger variability in the durations of the busy periods. For the same value of ϵ_K, a larger coding block size K helps with learning the PU spectrum. Figure 10 shows the region of the PU and SU throughputs, r_P and r_S, jointly achieved under the POMDP algorithm. For the same K, the PU can achieve higher throughput r_P for the same r_S and smaller ϵ_1 (and hence smaller ϵ_K). Also, for the same r_P and ϵ_1, r_S increases with increasing K; and for the same r_P and same K, r_S decreases with increasing ϵ_1 (and consequently with increasing ϵ_K). This is the combined effect of spectrum availability and spectrum predictability gains. Note that the maximum value of $r_S(= 1 - u(K))$ increases with increasing K due to the spectrum availability gain, and for $K = 2$ and 3, it corresponds to a non-zero r_P due to the spectrum predictability gain which allows to the PU collision free slots during every busy period as previously explained in Sec. III-B.

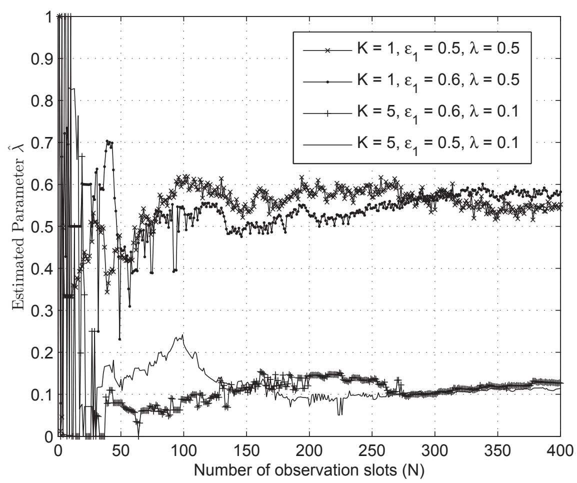

Fig. 9: Estimated $\hat{\lambda}$ vs. N under the erasure model.

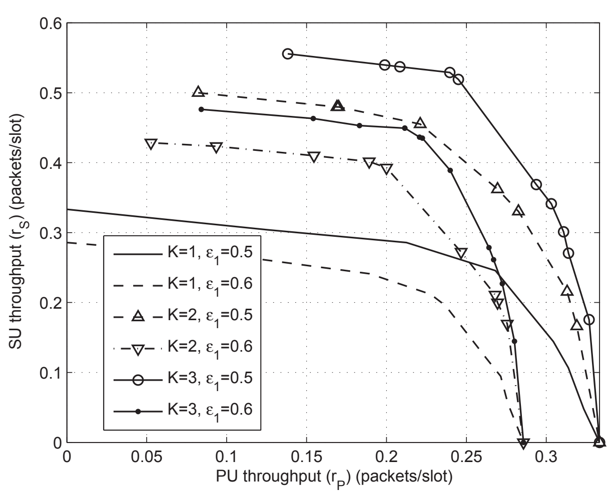

Fig. 10: Throughput performance of the POMDP algorithm with erasures.

V. Conclusion

We showed how the SU can leverage the structure of the idle/busy periods on the PU spectrum with network-coded transmissions. We first considered a perfect PU channel without any spectrum availability gain due to network coding, but even in this case the SU can largely benefit from improved spectrum predictability due to the structure induced by network coding on the PU spectrum. The Baum-Welch algorithm is applied to estimate the parameters of the PU's Markov chain. Once the PU Markov chain is known to the SU, we considered the SU's objective of maximizing its throughput and the POMDP algorithm is applied for tracking the PU's spectrum state evolution. For both cases (learning and tracking), we showed that increasing the coding block size is always beneficial for the SU. If the PU channel is not perfect, network coding also helps reduce the channel utilization needed by the PU to achieve the same throughput, leaving more idle slots for the SU to access the spectrum. In this case, we showed how the SU throughput improves with the aggregate gain of spectrum predictability and availability when the PU uses network coding for its transmissions.

References

[1] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks (Elsevier) Journal*, vol. 50, no. 4, pp. 2127–2159, September 2006.

[2] T. Yücek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, First Quarter 2009.

[3] Y. Chen, Q. Zhao, and A. Swami, "Joint design and separation principle for opportunistic spectrum access in the presence of sensing errors," *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 2053–2071, May 2008.

[4] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized cognitive Mac for opportunistic spectrum access in ad hoc networks: A POMDP framework," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 3, April 2007.

[5] R. Ahlswede, N. Cai, S. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[6] T. Ho, R. Koetter, M. Medard, M. Effros, J. Shi, and D. Karger, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, October 2006.

[7] A. Eryilmaz, A. Ozdaglar, and M. Médard, "On the delay and throughput gains of coding in unreliable networks," *IEEE Transactions on Information Theory*, vol. 54, no. 12, pp. 5511–5524, December 2008.

[8] N. Baldo, A. Asterjadhi, and M. Zorzi, "Dynamic spectrum access using a network coded cognitive control channel," *IEEE Transactions on Wireless Communications*, vol. 9, no. 8, pp. 2575–2587, August 2010.

[9] S. Wang, Y. E. Sagduyu, J. Zhang, and J. H. Li, "The impact of induced spectrum predictability via wireless network coding," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 2, pp. 758–769, February 2012.

[10] C. Ghosh, C. Cordeiro, D. P. Agrawal, and M. B. Rao, "Markov chain existence and Hidden Markov models in spectrum sensing," in *IEEE International Conference on Pervasive Computing and Communications (PerCom), 2009*, March 2009.

[11] L. R. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[12] R. D. Smallwood and E. J. Sondik, "The optimal control of partially observable Markov processes over a finite horizon," *Operations Research*, vol. 21, no. 5, pp. 1071–1088, 1973.

[13] C. H. Papadimitrou and J. N. Tsitsiklis, "The complexity of Markov decision processes," *Math. Oper. Res.*, vol. 12, no. 3, pp. 441–450, 1987.