Reinforcement Learning-based Cooperative Sensing in Cognitive Radio Ad Hoc Networks

Brandon F. Lo and **Ian F. Akyildiz**
Broadband Wireless Networking Laboratory, School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332
Email: {brandon.lo,ian}@ece.gatech.edu

***Abstract*—In cognitive radio networks, spectrum sensing is a fundamental function for detecting the presence of primary users in licensed frequency bands. Due to multipath fading and shadowing, the performance of detection may be considerably compromised. To improve the detection probability, cooperative sensing is an effective approach for secondary users to cooperate and combat channel impairments. This approach, however, incurs overhead such as sensing delay for reporting local decisions and the increase of control traffic in the network. In this paper, a reinforcement learning-based cooperative sensing method is proposed to address the cooperation overhead problem. By using the proposed cooperative sensing model, the secondary user learns to (i) find the optimal set of cooperating neighbors with minimum control traffic, (ii) minimize the overall cooperative sensing delay, (iii) select independent users for cooperation under correlated shadowing, and (iv) improve the energy efficiency for cooperative sensing. The simulation results show that the proposed reinforcement learning-based cooperative sensing method reduces the overhead of cooperative sensing while effectively improving the detection performance to combat correlated shadowing.**

I. Introduction

The fundamental cognitive radio (CR) function for primary user (PU) detection and spectrum exploration in licensed bands is spectrum sensing in CR networks [1], [2]. The primary goal of spectrum sensing is to identify available spectrum for secondary user (SU) transmission while reducing the interference with PUs to a tolerable level. However, due to the shadowing effects and multipath fading in wireless channels, the sensing results obtained by an individual SU is susceptible to detection errors. Imperfect sensing results in either wasting spectrum opportunity, known as a false alarm when a PU is mistakenly considered present in an available spectrum, or interfering with PUs, known as a miss detection when a present PU in the licensed band is wrongfully considered absent. To resolve this issue, previous studies [3], [4] demonstrated that cooperative spectrum sensing can effectively combat shadowing and multipath fading. Since the probability of all cooperating SUs in deep fading or shadowing is small, combining local sensing results to make detection decisions for all participating SUs reduces the possibility of making incorrect decisions at each individual SU. Thus, the cooperation among SUs for PU detection is a common and effective approach to combat shadowing and fading and to reduce the miss detection and false alarm probabilities.

Regardless of the benefits of cooperative sensing, cooperation incurs overhead that limits the cooperative gain. There are several major types of overhead limiting the gain obtained from cooperation: (i) the control message overhead, (ii) the delay incurred by reporting sensing results, (iii) correlated observations due to spatially correlated shadowing, and (iv) energy inefficiency. First, SUs exchange control messages on a common control channel to facilitate cooperation. To minimize the amount of control messages and the requirement of control channel bandwidth, the sensing results may be quantized before being reported. For example, sending a binary hard decision from local sensors is one way of reducing the control overhead. Second, synchronizing SUs in CR ad hoc networks for sensing cooperation is not a trivial task. Since each SU has different transmission and sensing schedules, the local sensing results from cooperating SUs may not simultaneously arrive at the SU making the cooperative decision. Moreover, the delay due to packet collision and re-transmission in control channel must be considered in the modeling.

In addition to control message overhead and delay problems, spatially correlated shadowing poses a limit on the achievable cooperative gain. Due to correlated shadowing, the observations at cooperating SUs may not be independent. As a result, the assumption of conditional independence in likelihood ratio testing (LRT) schemes [10], [12] does not hold in this case. Finally, energy spent on cooperative sensing is inefficient if correlated SUs perform local sensing and transmit the results that may compromise the accuracy of cooperative sensing detection. Therefore, to address all these problems, a new model for cooperative spectrum sensing is required in CR ad hoc networks (CRAHNs).

In this paper, we introduce a new model based on the reinforcement learning for cooperative sensing in CRAHNs to address the problems of associated overhead in correlated shadowing environment. Reinforcement learning (RL) [9] is an adaptive method for a decision-making agent learning to choose optimal actions and maximize the received reward by interacting with its environment. In the context of cooperative sensing, the SU initiating the cooperative sensing and making global detection decisions is the decision-making agent (simply called *agent* thereinafter) who learns the behaviors of cooperating neighbors and takes actions to improve the spectrum sensing results. To the best of our knowledge, this is the first work applying the techniques of reinforcement learning to address both the cooperation overhead problem and the detection performance of cooperative sensing.

Our contribution can be summarized as follows:

- We propose a novel cooperative spectrum sensing model based on reinforcement learning for SUs to learn the optimal set of cooperating neighbors and their report sequence while minimizing total cooperative sensing delays and energy inefficiency.
- We show that the optimal solution by RL-based approach greatly improves the detection performance under

978-1-4244-8016-6/10/$26.00 ©2010 IEEE

correlated shadowing while minimizing control channel bandwidth requirement by using binary local decisions and hard-combining strategy.

The remainder of this paper is organized as follows: Section II discusses the system model for cooperative sensing. Section III describes the proposed RL-based cooperative sensing model and algorithm. Section IV evaluates the performance by test scenarios, and finally, Section V concludes the paper.

II. System Model

We consider a group of SUs forming a CRAHN. As shown in Fig. 1, the network is overlaid with a primary network where more than one PU may appear in a set of licensed channels. The range of PU transmissions forms the protected region. Due to fading, shadowing, and the hidden primary receiver problem, a SU in the CRAHN may be unable to determine the availability of a particular licensed channel by local sensing. In this case, the SU broadcasts the request for cooperative sensing to all its one-hop neighbors. Selected cooperating neighbors respond to this request by performing local sensing and returning their local decisions *asynchronously*. The SU initiating the cooperative sensing process combines these local results and makes a detection decision on the presence of the PU. Thus, in our model, we focus on the cooperative sensing by one SU (the agent) and its one-hop neighbors on one licensed channel.

As shown in Fig. 2, the agent, also known as fusion center, initiates RL-based cooperative sensing and combines the local decisions. We assume that the agent has L one-hop neighbors. For each round of cooperative sensing, only $K \leq L$ neighbors are selected for cooperation. In addition, a reliable narrowband common control channel is used by the agent for broadcast and by cooperating SUs to report local sensing decisions back to the agent. We also assume that a channel is unavailable for SUs when at least one PU occupies the channel. Without loss of generality, only one PU in each licensed channel is considered. The PU activity is modeled as a two-state birth-death process with the birth rate r_b and the death rate r_d [6]. In this PU model, the transitions follow a Poisson process with exponentially distributed inter-arrival time. Thus, the long-term average probability of PU active (P_{on}) and inactive (P_{off}) are $r_b/(r_b + r_d)$ and $r_d/(r_b + r_d)$, respectively.

Let y_i be the average signal-to-noise ratio (SNR) of the received PU signal observed at cooperating user $i, i = 1, \ldots, L$. The observations may be correlated depending on the location of the SUs. The collection of these observations is the Gaussian distributed vector $\mathbf{Y} = \{y_i\}_1^L$ under the null hypothesis H_0, which indicates the absence of the PU transmit signal, and the alternative hypothesis H_1, which indicates the presence of the transmit signal, as follows [10]:

$$\mathbf{Y} \sim \begin{cases} \mathcal{N}(\mathbf{0}, \sigma_0^2 \boldsymbol{I}), & H_0 \\ \mathcal{N}(\boldsymbol{\mu_1}, \sigma_1^2 \boldsymbol{\Sigma}), & H_1 \end{cases} \tag{1}$$

where $\mathbf{0}$ is the zero vector, $\boldsymbol{\mu_1}$ is the mean SNR that depends on the path loss from the location of the PU, σ_0^2 is the Gaussian noise variance under H_0, σ_0^2 is the variance of noise and correlated shadowing under H_1, $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{\Sigma}$ is the normalized covariance with elements ρ_{ij}. We assume that the correlation follows the exponential correlation model [5]. In this model, $\rho_{ij} = e^{-ad_{ij}}$, where a is the exponential decaying coefficient set to 0.1204 and 0.002 for urban and suburban settings, respectively [4], and d_{ij} is the distance between SUs i and j.

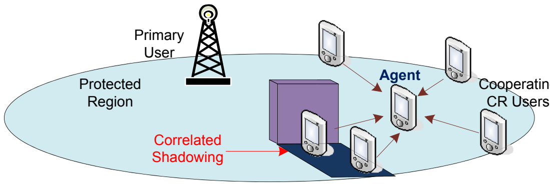


Fig. 1. RL-based cooperative sensing under correlated shadowing.

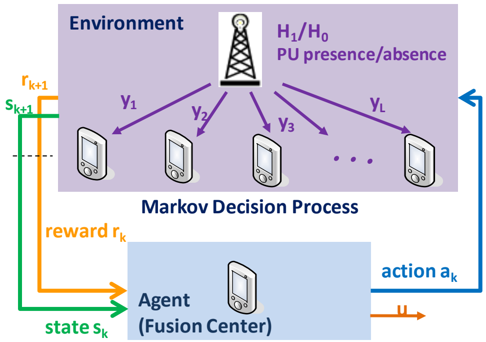


Fig. 2. Reinforcement learning model for cooperative sensing.

Based on the independent observations obtained locally, each selected SU makes a local decision with the detection threshold λ_{th}. Without the closed-form expression, the detection threshold, same for all SUs, is determined by the simulations [10], [11]. Depending on the report delay of each neighbor and the actions selected by the agent, the cooperating SUs report the binary decision: “1” and ”0” indicate the presence and absence, respectively, of the PU on the licensed channel. After collecting the results of selected SUs for each round of cooperative sensing, the agent uses the counting and majority rules to determine the cooperative decision on the presence of the PU. The decision is then broadcast to all neighbors.

For the agent to learn from the environment and find the optimal solution, the aforementioned cooperative sensing process is repeated for N iterations, called *episodes*. Thus, the learning time is the summation of the cooperative sensing time in each episode.

III. RL-based Cooperative Sensing

In this section, we present the model and the algorithm for the RL-based cooperative sensing. First, we formulate the cooperative sensing problem as a finite-horizon Markov decision process (MDP) [7]. The reasons of using the MDP model are twofold: (i) the agent makes sequential decisions on selecting cooperating neighbors and (ii) the optimal solution can be obtained by reinforcement learning algorithms in which inherently consist of an MDP. Our objective is to find the optimal set of cooperating neighbors, minimize their report delay, and improve the detection performance under correlated shadowing. Thus, we introduce the MDP model for the cooperative sensing problem and the algorithm to find the optimal solution.

A. Finite-Horizon Markov Decision Process Model

In our MDP model, the agent interacts with the cooperating neighboring nodes that observe the PU activity in the environment. It then receives a response from a neighbor as the reward for next state and the selection of next action. By exploring the unknown states and accumulating the knowledge of receiving rewards from known states, the agent learns the sequence of optimal decisions that gives rise to the maximum reward. Fig. 2 also illustrates this inherent MDP. The MDP and agent's two major tasks during the learning process: action selection and reward calculation are discussed next.

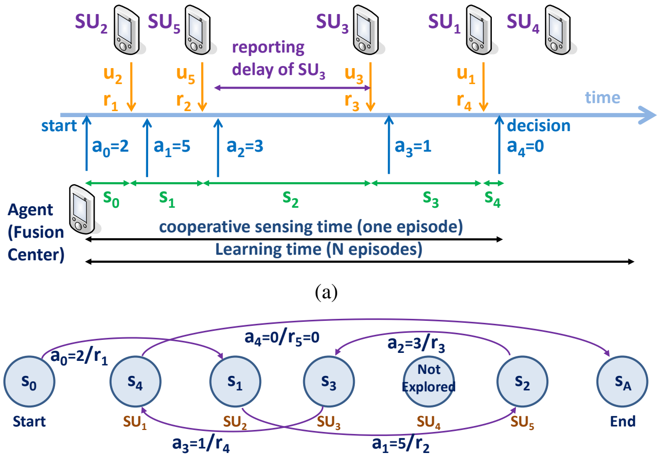
1) MDP: The finite-horizon MDP is represented by a quadruple $\langle \mathcal{S}, \mathcal{A}, f_p, f_R \rangle$ in which $\mathcal{S}$ is a finite set of all states, $\mathcal{A}$ is a finite set of all possible actions, f_p is the state transition probability function, and f_R is the reward function. Since the state transition probability is not known a priori and not required, each component except f_p is described as follows:

States: Let L be the number of all cooperating SUs. In each state $s_k \in \mathcal{S}, 0 \leq k \leq (L+1)$, the agent chooses an action a_k based on the action selection strategy and awaits a response from the environment. The response is a quantized local decision u_i transmitted by a cooperating neighbor SU_i with report delay t_d. The report delay and estimated correlation are used for reward calculation. The state changes from s_k to s_{k+1} when the agent obtains the reward r_{k+1}. The number of states is $|\mathcal{S}| = L+2$. Those two extra states are *start* and *end*.

Actions: An action is a decision an agent makes in a state. The agent in state s_k chooses an action $a_k = i \in \mathcal{A}, 0 \leq i \leq L$ in which i indicates the ID i of the selected SU and L is the number of cooperating SUs. Thus, the agent requests the local decision u_i from neighbor SU_i by sending action $a_k = i$, $1 \leq i \leq L$. When the agent decides to terminate the cooperation process and make a cooperative decision, it informs all cooperating SUs by action $a_k = 0$. Nevertheless, how to choose the actions depends on the action selection strategy. Note that both the number of states and the number of actions grow *linearly* with the number of cooperating SUs.

Reward function: The reward function maps the state-action-state transition to a real-valued reward as $f_p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$. The reward is used for evaluating the choice of current action and the performance of the entire learning process. Thus, the agent calculates a reward r_{k+1} upon the arrival of a local decision u_i from SU_i as a result of action $a_k = i$ in state s_k.

Fig. 3(a) gives an example of the MDP. The agent initially starts in state s_0, the *start* state. After choosing the first action $a_0 = 2$, it waits for the first reward r_1. In the figure, the agent's action is asking the neighbor SU_2 to report its local sensing result. Upon the receipt of the reward r_1 from SU_2, the state changes to the next state s_1. Then the agent chooses SU_5 with action $a_1 = 5$. This cycle continues from state s_1 to state s_4. In s_4, the agent chooses the action $a_4 = 0$ to terminate the cooperative sensing process by entering the *end* state s_A. Thus, SU_4 is not selected in this case. Fig. 3(b) illustrates the state diagram for the corresponding state transitions. The state transition from s_k to s_{k+1} occurs when the agent takes action $a_k = i$ and receives reward r_{k+1} in state s_k. Thus, in each state, the agent either awaits the next SU to respond after taking a new action or decides to end the process.



(a)

(b)

Fig. 3. (a) The inherent Markov decision process and (b) the corresponding state diagram in the reinforcement learning model.

2) Action Selection Strategy: We use a softmax approach based on Boltzmann distribution for action selections. In this action strategy, the probability of selecting action a_i in state s_k is given by:

$$p(s_k, a_k = i) = \frac{e^{Q(s_k, a_k = i)/\tau_n}}{\sum_{j=1}^{N_a} e^{Q(s_k, a_j)/\tau_n}}, \qquad i = 1, \ldots, N_a \quad (2)$$

where $Q(s_k, a_k = i)$ is the state-action value function that evaluates the quality of choosing action $a_k = i$ in state s_k, N_a is the number of actions, $|\mathcal{A}|$, and τ_n is a time-varying parameter called *temperature* that controls the degree of exploration versus exploitation. For large value of τ_n, all actions are equally probable. In this case, the agent explores the opportunities of more uncorrelated cooperating SUs to achieve potentially higher detection probability in the future with large τ_n. On the other hand, for small τ_n, the action with maximum $Q(s, a)$ is favored. Hence, the agent exploits the current knowledge of best selections of cooperating SUs to achieve the potentially highest detection probability with small τ_n. As more episodes are devoted to sensing that changes the strategy from exploration toward exploitation, the value changes from large to small to assure that the convergence is achieved [8]. Thus, we use a linear function to decrease the value of τ_n over episodes as follows: $\tau_n = -(\tau_0 - \tau_N) \cdot n/N + \tau_0$, where N is the total number of episodes, τ_0 and τ_N are the initial and the last value of the temperature, respectively.

3) Cumulative Reward Calculation: The reward is a function of the reporting delay and the correlation incurred by the newly selected SU_i. The reporting delay $t_d(s_k, a_k = i)$ is the time between the time of the agent requesting SU_i's cooperation with the action $a_k = i$ and the arrival time of the local decision u_i at the agent. The delay is characterized by the delay cost factor C_d given by:

$$C_d = \frac{\sum_{j=0}^{k-1} t_d(s_j, a_j) + t_d(s_k, a_k = i)}{\sum_{j=0}^{L-1} t_d(s_j, a_j)}. \quad (3)$$

It is simply the fraction of cumulative reporting time up to the next state s_{k+1} to the total reporting time for all neighbors. On the other hand, the cost associated with correlation is the cumulative correlation coefficients between the newly selected SU_i and all selected SUs in previous k states. Given SU_j selected in state $s_m, m = 0, \ldots k-1$ and the correlation coefficient matrix $\mathbf{\Sigma} = \{\rho_{ij}\}$ estimated from the location information, the cost can be expressed by:

$$C_\rho = \sum_{m=0}^{k-1} \rho_{ji}(s_m, a_m = j) \tag{4}$$

with SU_i selected in state s_k. Thus, the new reward r_{k+1} responding to the action a_k in state s_k observed by the agent is given by:

$$r_{k+1}(s_k, a_k) = \begin{cases} 1 - C_d, & C_\rho = 0 \\ -C_\rho, & C_\rho \neq 0 \end{cases} . \tag{5}$$

The cumulative reward for episode n is $R_n = \sum_{i=1}^{K} r_i$, where K is the number of SUs selected in this episode. Therefore, the ultimate goal is to maximize the expected cumulative reward averaged over the most recent N_{epi} episodes given by: $R^{\pi^*} = E[R_n^*] = E\big[\sum_{i=1}^{K^*} r_i\big]$, where K^* is the optimal number of cooperating SUs and π^* is the optimal policy mapping the state s_k to the action a_k for $0 \leq k \leq K^*$ to achieve the maximum cumulative reward.

B. RL-based Cooperative Sensing Algorithm

The RL-based cooperative sensing algorithm is listed in Algorithm 1. The algorithm takes the array of all state-action values $(Q(|\mathcal{S}|, |\mathcal{A}|))$ as the input and initializes the array entries to zero. The output is the optimal solution: optimal number of cooperating SUs (K^*), optimal sensing sequence ($\mathcal{P}^*$), and minimum reporting delay (t_d^*). In the algorithm, RL-based cooperative sensing is performed for N episodes (lines 4 to 18). In each state of an episode, a local decision is received by sending the request to the selected neighbor, which is selected by the action strategy. At the end of the episode, a cooperative sensing decision is determined.

Before each state changes, the state-action values are updated by the equation in line 10 in which $Q_k = Q(s_k, a_k)$, α_k is the learning rate, γ is the discount factor of future state-action value estimates Q_{k+1}, and $f(Q_{k+1})$ is the function of future estimates Q_{k+1}. The learning rate α_k is adjusted each state such that the learning is slower for those state-action pairs explored often and faster for those less frequently explored. $f(Q_{k+1})$ depends on the temporal-difference (TD) learning algorithms [9] used.

At the end of the reinforcement learning process, the optimal solution is obtained by searching for the sequence with the maximum state-action value in each state of the state-action array (lines 20 to 27).

Algorithm 1 : RL-based Cooperative Sensing

```
 1: Input: $L, N, Q$
 2: Output: $K^*, \mathcal{P}^*, t_d^*$
 3: Initialize$(Q(|\mathcal{S}|, |\mathcal{A}|))$
 4: for $n \leftarrow 1$ to $N$ do
 5:     for $k \leftarrow 0$ to $(L+1)$ do
 6:         $a_k \leftarrow$ ActionStrategy$(s_k, Q, \tau_n)$
 7:         if $a_k \neq 0$ then
 8:             SendReqToNeighbor$(SU_i; a_k = i)$
 9:             $r_{k+1} \leftarrow$ WaitForReward$(u_i; t_d(s_k, a_k = i))$
10:             $Q_k \leftarrow (1 - \alpha_k)Q_k + \alpha_k\big[r_{k+1} + \gamma f(Q_{k+1})\big]$
11:             UpdateLearningParameters$(\alpha_k, \gamma, \tau_n)$
12:         else
13:             break
14:         end if
15:     end for
16:     $u \leftarrow$ CoopSensingDecision$(u_i, k, L)$
17:     BroadcastCoopDecision$(u; a_k = 0)$
18: end for
19: $k, j, t_d^* \leftarrow 0; \mathcal{P}^* \leftarrow \emptyset$
20: repeat
21:     $i \leftarrow \arg\max_j Q(k, j)$
22:     if $Q(k, i) \neq 0$ then
23:         $\mathcal{P}^* \leftarrow \mathcal{P}^* \cup \{SU_i\}$
24:         $t_d^* \leftarrow t_d^* + t_d(s = k, a = i)$
25:         $k \leftarrow i; K^* \leftarrow |\mathcal{P}^*|$
26:     end if
27: until $Q(k, i) == 0$
```

IV. Performance Evaluation

In this section, we evaluate the performance of our proposed scheme by finding the optimal solution, adapting to the environmental change, and improving the detection probability. This also shows the convergence and the adaptability of RL-based cooperative sensing. The optimal solution obtained by the RL-based cooperative sensing includes the optimal selection of cooperating SUs, the corresponding sensing sequence, and the minimum sensing delay.

We consider a SU (an agent) and its 9 neighbors deployed in a 600m × 600m square area $\mathfrak{A}$. Assume the square area is placed in the first quadrant of the Cartesian coordinate system. A PU who may be active or inactive on the 900MHz channel is located at the origin $(0, 0)$ (left bottom corner of $\mathfrak{A}$). The agent is located at $(500, 500)$ and its neighbors are located within 15 meters of the agent location. With this deployment, these SUs are approximately located at the border of the protected region of the PU. At this border location, the received power is close to the noise floor set to -101dBm. Thus, cooperative sensing is essential for them to improve their detection performance.

The birth rate r_b and the death rate r_d of the PU activity model are set to 0.3 and 0.2, respectively. Thus, the corresponding P_{on} and P_{off} are 0.6 and 0.4, respectively. When the PU is active, the PU transmit power is decayed by the path loss with path loss exponent $\gamma_{pl} = 3.1$. With the average received SNR after path loss as the mean, the received power under correlated shadowing varies with the shadowing spread $\sigma_1 = 6$dB. The decaying coefficient of the exponential correlation model is set to 0.1204 for urban settings. Based on this model, two SUs with a distance approximately larger than 8.3m apart are considered independent. When PU is inactive, the SUs observe the Gaussian noise with $\sigma_0 = 3$dB. In addition, the local detection threshold λ_{th} is set to 0.2dB for all SUs.

Table I lists one of random deployments of an agent (ID 0) and 9 cooperating SUs. The cooperating SUs are randomly deployed around the agent. The PU not shown in the Table is located at the origin $(0, 0)$. From their coordinates, one can find that SUs 1, 7, and 8 are strongly correlated. SUs 6 and 9 are correlated as well as SUs 3 and 5. In addition, each cooperating SU has different sensing report delay and sensing schedule, and may change its location over time. For illustration purpose, the location, sensing report delay, and

TABLE I
Location, Report Delays, and Schedule Priority of CR Users

ID	x-coord	y-coord	Delay (ms)	Priority
0	500	500	-	-
1	485	514	11.38	8
2	502	494	4.67	7
3	504	512	2.72	4
4	489	486	14.81	2
5	506	514	12.80	3
6	501	532	3.15	1
7	486	515	4.60	9
8	487	513	4.24	6
9	503	534	20.99	5

schedule priority of these SUs are assumed to be fixed in this scenario. From the table, the order of SU sensing report delay is $\{3, 6, 8, 7, 2, 1, 5, 4, 9\}$ while the schedule priority is $\{6, 4, 5, 3, 9, 8, 2, 1, 7\}$. For example, SU_3 has the smallest sensing delay among all cooperating SUs and SU_6, if selected, is the first SU to report local decisions.

In this test scenario, the RL-based cooperative sensing is performed by using Q-learning, Sarsa, and Actor-Critic algorithms [9] for $N = 5000$ episodes. The learning rate is updated by $\alpha_k = 1/(1 + f_v(s_k, a_k))$, where $f_v(s_k, a_k)$ is the frequency of exploring the (s_k, a_k)-pair and is incremented each time (s_k, a_k) is explored, and the discount factor is set to 0.9. With different initial temperature τ_0 values and the same final value $\tau_{5000} = 0.01$, all three methods converge at episode 5000 and reach the optimal solution: the optimal set of cooperating SUs in the sensing report order is $\{6, 4, 3, 8, 2\}$ with the total sensing report delay 29.58ms. Compared to the full cooperation by all 9 neighbors, which requires total delay 79.35ms, a 63% saving of cooperative sensing time is obtained in this case. It is also more energy-efficient (up to 44% saving) since only selected neighbors (optimal $5 <$ total 9 SUs) are required to perform local sensing and report their results for each round of cooperation. Thus, the detection performance, sensing delay, and energy efficiency of RL-based cooperative sensing are considerably improved from the cooperation by all neighbors under correlated shadowing.

The performance of obtaining the optimal solution is further evaluated by the expected cumulative rewards, learning time, and detection performance, which are discussed next.

1) Expected Cumulative Rewards: Since the expected cumulative reward is negatively proportional to the sensing delay and correlation, higher reward represents the selection of more independent SUs with smaller sensing delay. However, since the reward observed at each state is bounded and the number of states is finite for each episode, the expected cumulative reward asymptotically reaches its upper bound when the algorithm converges.

Fig. 4 shows the expected cumulative rewards averaged over the most recent 100 episodes ($N_{epi} = 100$) for total 5000 episodes obtained by aforementioned three TD-learning methods. All three methods are converged to the optimal solution at episode 5000. However, with different τ_0, each algorithm requires different exploration and exploitation strategy to reach the optimal solution. Q-learning with $\tau_0 = 1$ tends to explore during the first 2500 episodes and exploit during the second half. As the result, the expected cumulative reward of Q-learning is constantly improving until the highest reward 3.768 is reached long before the final episode. Sarsa with $\tau_0 = 0.5$ tends to exploit the immediate rewards more in the early stage of learning. As a result, the expected cumulative reward of Sarsa is higher than that of Q-learning during the first half, but fails to match that of Q-learning in the second half. The Actor-critic method with $\tau_0 = 100$, on the other hand, requires long exploration period to reach the optimal solution. Its expected cumulative reward fails to match that of previous two methods for most of the learning period. From these observations, Q-learning exhibits the balance of exploration and exploitation and constant improvement in the expected cumulative rewards. Thus, we mainly focus on Q-learning for the rest of discussion.

2) Learning Time: In cooperative sensing, total cooperative sensing time must be minimized to improve the throughput of SUs. To reduce time required for the learning process, the RL-based algorithm must be able to converge at faster speed or settle on an acceptable suboptimal solution. In our test scenario, we reduce the number of episodes n from 5000 to 500 with the $\tau_0 = 1$ and $\tau_n = 0.01$. In other words, the τ_n value is changed at higher rate with smaller number of episodes to run. Note that the average number of exploration in each state is linearly decreased with the number of episodes. This means that when the number of episodes gets smaller, we take more risks of settling on a suboptimal solution due to the lack of exploration at certain point.

Fig. 5 shows the expected cumulative rewards for number of episodes 500, 800, 2500, and 5000. In fact, the optimal solution can be obtained with the number of episodes from 800 to 5000 as the maximum reward can be reached in each case. However, as the case of $n = 500$ shows, the algorithm will settle on a suboptimal solution when the number of episodes is less than 800. Thus, the RL-based cooperative sensing with Q-learning provides the agent the flexibility to make tradeoffs between longer learning for more stable detection performance and shorter sensing time for larger throughput. By reasonably reducing the learning time, the agent will be able to reach the optimal solution with high probability as well as minimizing the sensing time for more transmission opportunities.

3) Detection Performance: In addition to the sensing delay and selection of independent cooperating SUs, the improvement in detection performance under correlated shadowing is the primary focus of our proposed scheme. After the end of each episode, the agent combines the local decisions and makes a cooperative sensing decision. Even before the optimal solution is reached, the detection performance can be considerably improved as the RL-based cooperative sensing is in progress.

Fig. 6 shows the improvement of detection performance during the reinforcement learning process. In this figure, "full coop" represents the cooperation from all neighboring SUs while "RL coop" represents the cooperation from the SUs selected by the agent based on the RL-based cooperative sensing. The probabilities of detection and false alarm are calculated based on the PU activity and the most recent 500 sensing decisions made by the agent. With the same setting as in Section IV-1), it is evident that the detection probability (P_d) is gradually improved and reaches above 0.9 after 4200 episodes. The false alarm probability (P_f) is considerably reduced to 0.1. Thus, with RL-based cooperative sensing, the detection performance improves as soon as the learning from the environment takes effects.

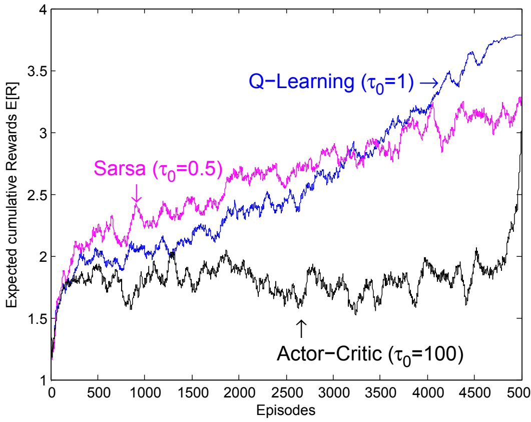


Fig. 4. Expected cumulative rewards of RL-based cooperative sensing.

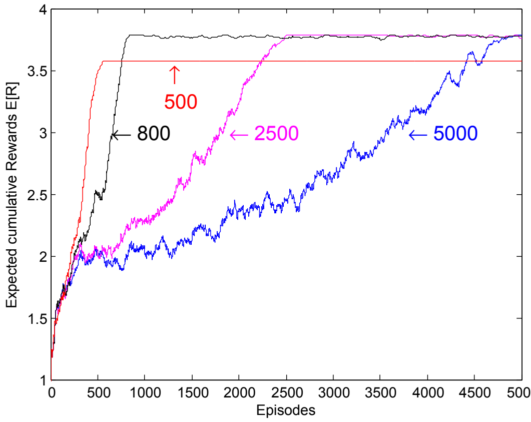


Fig. 5. Performance comparison in reducing the learning time.

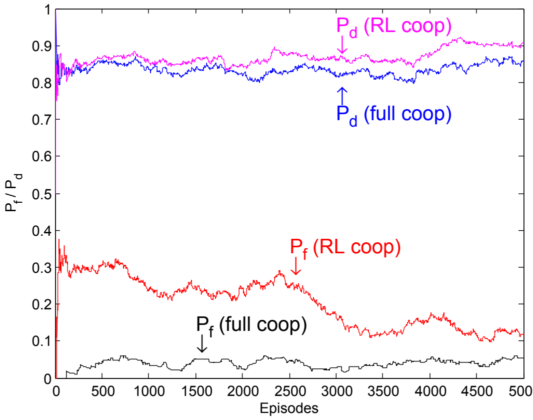


Fig. 6. Improvement of P_d and P_f during RL-based cooperative sensing.

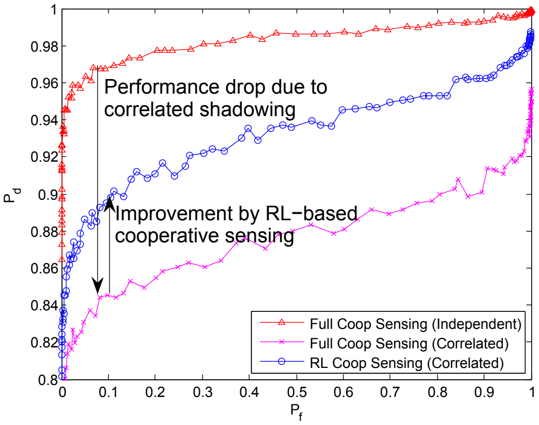


Fig. 7. ROC of RL-based cooperative sensing and full cooperation cases.

When the optimal solution is obtained at the episode 5000, the detection probability P_d is improved to over 0.9 while the false alarm probability P_f is reduced to approximately 0.1. Fig. 7 shows the receiver operating characteristic (ROC) curve of the RL-based cooperative sensing with the optimal set of cooperating users. The ROC curves of full cooperations by all neighbors under independent and correlated shadowing are included for comparison. If P_f is set to 0.1, P_d drops from 0.97 to 0.85 under correlated shadowing. With RL-based cooperative sensing, P_d is increased to 0.90. Therefore, our proposed scheme is effective in combating correlated shadowing.

V. Conclusions

In this paper, we propose a novel cooperative sensing model based on reinforcement learning to minimize the cooperation overhead and improve the detection performance under correlated shadowing. With several temporal-difference learning methods under consideration, RL-based sensing with Q-learning exhibits the best performance of making the balanced tradeoffs between exploration and exploitation in our model. We also show that the proposed RL-based scheme is capable of converging to the optimal solution and adapting to the environment change. Due to its learning capability, this work is extended to consider the movement of SUs, the change of PU models, and the data falsification problem such as learning the malfunctioning of SUs and the attacks of malicious users that compromise the accuracy of cooperative sensing detection.

References

[1] I. F. Akyildiz, W.-Y. Lee, and K. R. Chowdhury, “CRAHNs: cognitive radio ad hoc networks,” *Ad Hoc Networks Journal (Elsevier)*, vol. 7, pp. 810–836, Jul. 2009.

[2] 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 Journal (Elsevier)*, vol. 50, pp. 2127–2159, Sept. 2006.

[3] D. Cabric, S. M. Mishra, and R. W. Brodersen, “Implementation issues in spectrum sensing for cognitive radios,” in *Proc. of 38th Asilomar Conf. on Signals, Systems, and Computers*, Nov. 2004, pp. 772–776.

[4] A. Ghasemi and E. S. Sousa, “Collaborative spectrum sensing for opportunistic access in fading environments,” in *Proc. of IEEE DySPAN 2005*, Nov. 2005, pp. 131–136.

[5] M. Gudmundson, “Correlation model for shadowing fading in mobile radio systems,” *Electronics Letter*, vol. 27, no. 23, pp. 2145–2146, Nov. 1991.

[6] W.-Y. Lee and I. F. Akyildiz, “Optimal spectrum sensing framework for cognitive radio networks,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 10, pp. 3845–3857, Oct. 2008.

[7] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY: John Wiley & Sons, 1994.

[8] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári, “Convergence results for single-step on-policy reinforcement learning algorithms,” *Machine Learning*, vol. 38, no. 3, pp. 287–308, Mar. 2000.

[9] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: The MIT Press, 1988.

[10] J. Unnikrishnan and V. V. Veeravalli, “Cooperative sensing for primary detection in cognitive radio,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 18–27, Feb. 2008.

[11] E. Visotsky, S. Kuffner, and R. Peterson, “On collaborative detection of TV transmissions in support of dynamic spectrum sharing,” in *Proc. of IEEE DySPAN 2005*, Nov. 2005, pp. 338–345.

[12] F. E. Visser, G. J. Janssen, and P. Pawelczak, “Multinode spectrum sensing based on energy detection for dynamic spectrum access,” in *Proc. of IEEE VTC-Spring 2008*, May 2008, pp. 1394–1398.