Monte Carlo Reinforcement Learning for Cooperative Spectrum Sensing in Decision Fusion

Qingying Wu, Benjamin K. Ng(✉), Han Zhu, and Chan-Tong Lam

Faculty of Applied Sciences, Macao Polytechnic University, Macao 999078, China
{qingying.wu,bng,ctlam}@mpu.edu.mo, HanZhu@ieee.org

Abstract. As one of the key enablers, the Wireless Sensor Network (WSN) plays an important role in wide application scenarios of the Internet of Things (IoT). However, the rapid spread of wireless applications contributed to the extreme crowd in the radio spectrum. Cognitive Radio Sensor Network (CRSN) emerges as a promising solution to the problem of spectrum scarcity considering the heterogeneous properties of both the Primary User (PU) and Secondary User (SU). In a multi-stage Cooperative Spectrum Sensing (CSS) system with a fusion center, hard fusion rules are widely used to fusion local decisions due to their simplicity. In this way, the sensing performance is closely related to the underlying parameters of the system but is hard to adjust when the fusion policy is fixed. This paper investigates the application of Monte Carlo Reinforcement Learning (MCRL) algorithms for CSS. Specifically, after replacing the traditional FC with a soft-created Agent, the policy for the fusion on local decisions can be improved intelligently using Monte Carlo Control while positively guiding the optimization of system performance. Experiments demonstrate that the proposed scheme can help achieve an ideal policy for better system performance in the global probabilities of detection and false alarm under various Signal-to-Noise Ratios (SNRs).

Keywords: Internet of Things · cognitive radio sensor networks · cooperative spectrum sensing · Monte Carlo Control

1 Introduction

Due to the quick spread of mobile devices with built-in sensors and processors, Wireless Sensor Network (WSN) has gained a lot of attention in both academia and industry [1,2]. A crucial requirement for WSN is how to deliver reliable real-time feedback using the existing wireless communication networks

This work was funded by the research funding of Macao Polytechnic University, Macau SAR, China (Project no. RP/ESCA-02/2021) and The Science and Technology Development Fund, Macau SAR, China (File no. 0044/2022/A1).

V. C. M. Leung et al. (Eds.): Qshine 2023, LNICST 574, pp. 211–225, 2024.
https://doi.org/10.1007/978-3-031-65123-6_16

with constrained spectrum resources [3]. To alleviate the typical spectrum shortage problem [4], spectrum sensing is a promising technology that can be introduced to WSN, which can detect the vacant spectra (Spectrum Hole, SH) that the Primary User (PU) unused for the Secondary User (unlicensed user, SU). Multiple wireless devices can be incorporated using this technology for a variety of applications, including the Internet of Things (IoT), smart grids, etc.

Cooperative Spectrum Sensing (CSS) cooperates with more than one SU together to assess PU's state with higher accuracy, and further cope with severe noise and fading. The entire CSS process, which is carried out by numerous cooperative sensors, can be divided into four steps [5]: (1) Single nodes' spectrum sensing, (2) Each node's reporting local sensing data, (3) Fusion all single node's result, and (4) Fusion Center (FC) broadcasting the fusion decision.

In contrast to the other two common techniques, which are, Cyclostationary Feature Detection and Matched Filter, the same popular technology called Energy Detection (ED) possesses wider application in local spectrum sensing for its easiest complexity and least computation. ED was originally designed for unknown definite signals and was first described in 1967 [6]. No prior knowledge is required in the detection process. As an efficient way of detecting signals, it accesses the result by comparing the total energy received during a given period of time to a predetermined threshold. Although the complexity of calculation and implementation can be considerably decreased, it's still incompetent when faced with uncertainty due to noise.

According to the way of sharing the sensing information, the local decision made by each node can be transmitted in binary or encoded, which are also known as hard-decision and soft-decision reporting schemes. Hard-decision combining ('OR' Rule, 'AND' Rule, and 'Majority' Rule) and Soft fusion rules (Square Law Selection, Maximal Ratio Combining, Square-Law Combining, and Selection Combining) are two corresponding categories of fusion form. A series of comparisons between soft-based and hard-based fusion rules. Even though soft-decision-based approaches show more robustness to the bit error probability [7], experiments implemented demonstrated that soft-based fusion rules may outperform a little bit than hard-based ones [8,9], which are typically accompanied by more energy usage and computation overhead [10,11]. Due to its less complexity, the hard fusion rules have a wider application, but the quality of spectrum sensing after the decision-making step is decided by the underlying parameters and cannot be optimized easily.

Under the background of the rapid rise of information industries, wireless communication technology is developing at an extremely rapid speed within wide application ranges [12,13]. Problems in spectrum shortage that cannot be addressed with traditional methods present in a more increasing frequency, accompanying higher timeliness requirements. As a branch of Machine Learning (ML), Reinforcement Learning (RL) is a quite powerful solution for sequential decision-making under an uncertain system [14], which is greatly similar to the complex wireless communication system. The introduction of RL-based

algorithms is promising to help improve the performance of wireless communication systems in various aspects.

In recent years, many researchers focus on solving the traditional problems in wireless communication systems utilizing emerging RL-related algorithms. After transforming the energy efficiency issue of CSS into looking for the smallest subset of sensors with an efficient topology, there has been a study that utilizes Q-learning to guide learning sensor selection algorithms for energy efficiency by integrating data from graph structures into a neural network [15]. Most existing researches, such as [16–18] focus on the macro improvement of the spectra utilization. As a result, our work is unique in the sense that we pay attention to the optimization of the spectrum sensing process at the bottom layer by optimizing the traditional hard fusion rules with RL algorithms.

In this paper, we proposed a novel way to utilize Monte Carlo Reinforcement Learning (MCRL) to improve the traditional hard fusion rules so that the CSS system performance can be optimized, which can also be scaled to more realistic radio environments. The primary contributions can be summarized as follows.

(1) After formulating the CSS process into an RL problem, the traditional FC was replaced with a softly-created Agent to make fusion decisions based on the local decisions made by SUs according to the ED results.
(2) To make up for the defects that the system performance cannot be adjusted due to the fixed underlying parameters under traditional methods, ϵ-greedy algorithms with various greedy rates are adopted to implement the Monte Carlo Control. In this way, the policy acted by the Agent can be improved for better system performance.
(3) In the simulated environment, the spectrum sensing process was displayed and evaluated. At the same time, the feasibility of utilizing greedy policies was verified so that the metric for CSS performance could be measured and improved.
(4) Compared with traditional methods, our proposed method can indeed help improve the originally uncontrollable system performance.

The paper is organized as below. The preliminaries of the CRSN are discussed in Sect. 2. Then, after modeling the CSS process into an RL problem, Sect. 3 introduces the basic theory of the MCRL and the proposed method. In Sect. 4, the parameters for simulating a realistic environment are detailed, as well as the implementation process and results analysis are presented. Lastly, the results and future work are summarized.

2 Preliminaries on Cognitive Radio Sensor Network

As described in Fig. 1, a CRSN for CSS is made up of multiple CRs/SUs and an FC. The formers are capable of sensing the spectrum for local decisions in time slots, and the latter is in charge of making global fusion. By considering the heterogeneous properties of both PU and SUs, we make an effort to investigate the CSS issue from a more realistic perspective.

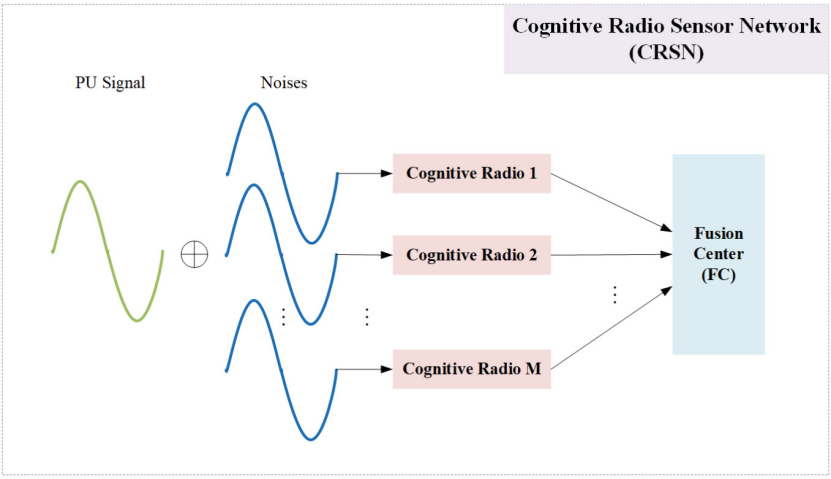


Fig. 1. Cognitive radio sensor network for cooperative spectrum sensing.

Usually, the PU channel is described from the channel capacity and the channel idle probability, whereas the SUs are reflected from the energy detection threshold, received Signal-to-Noise Ratio (SNR), and location geographically. In a CRSN of the flat-fading environment, suppose a total of M sensors, each of which collects N samples from the time slot $(i = 1, \cdots, N; j = 1, \cdots, M)$. For the monitoring of SHs and further identifying whether the PU is present (H_1) or absent (H_0), SUs always implement spectrum sensing with two hypotheses testing criteria as below. Specifically, denote the channel gain as h_j, the received signal at each sample of each sensor r_{ij} [19] is jointly composed of both the PU signal s_i and the Additive White Gaussian Noise (AWGN) ω_{ij}.

$$\begin{cases} H_1 : r_{ij} = h_j s_i + \omega_{ij}, \\ H_0 : r_{ij} = \omega_{ij}. \end{cases} \tag{1}$$

When considering ED as the spectrum sensing technology, the local spectrum sensing evaluates the signals' test statistics against thresholds, and further be compared to one or multiple predetermined threshold(s). The energy obtained at each sensor E_j can be expressed by

$$E_j = \sum_{i=1}^{N} \frac{r_{ij}^2}{\omega_{ij}^2} \tag{2}$$

If only one threshold is introduced, SUs can only declare $H_1\,(E_j \geq \lambda)$ or $H_0\,(E_j < \lambda)$, which can be impacted by unknown noise during real implementation [20]. Double thresholds can be utilized to avoid being affected by such issues [21], that is,

$$\begin{cases} send\,1,\ declaring\,H_1 & if\ E_j \geq \lambda_2, \\ no\,decision, & if\ \lambda_1 < E_j < \lambda_2, \\ send\,0,\ declaring\,H_0 & if\ E_j \leq \lambda_1. \end{cases} \tag{3}$$

The FC makes the fusion decision based on local decisions made by CR users according to the fusion rule (also known as the decision rule), which is a mechanism for determining licensed spectrum utilization in a CSS network. In this paper, we focus on the 3 hard fusion rules with fewer complexities as below.

1) 'OR' Rule ('1-out-of-N' Rule) [19,22], which specifies that the PU shall be deemed to be present in the FC if more than one node detects its presence.
2) 'AND' Rule ('N-out-of-N' Rule) [19,22], that is, the FC will determine that the PU does not exist unless all nodes confirm its existence.
3) 'Majority' Rule ('k-out-of-N' Rule) [23], i.e., if there are over k SUs reported for the PU's presence, then the PU is present. Typically, the 'Majority' rule's number of k equals half of the total SU number N, that is $N/2$.

To evaluate the reliability of CSS from a local and global aspect respectively, metrics including the local probabilities of detection $P_{d,j}$ and false alarm $P_{f,j}$ for j-th sensor expressed as

$$P_{d,j} = Pr\left(E_j \geq \lambda_2 \mid H_1\right) \tag{4}$$

$$P_{f,j} = Pr\left(E_j \geq \lambda_2 \mid H_0\right) \tag{5}$$

as well as the global probabilities of detection Q_d and false alarm Q_f defined as

$$Q_d = Pr\left(FD = 1 \mid H_1\right) \tag{6}$$

$$Q_f = Pr\left(FD = 1 \mid H_0\right) \tag{7}$$

are introduced. As discussed in [19,22], both Q_d and Q_f are based on the specific fusion rule, which will lead to different results in practical applications.

3 Monte Carlo Reinforcement Learning for Spectrum Sensing Improvement

According to conventional thinking, if the settings on sensors (Signal to Noise Ratio, SNR γ_j and sleeping rate μ_j for any $j = 1, \cdots, M$) and detection thresholds (λ_1 and λ_2) are known, PU's state can be obtained by the estimation based on ED technology and traditional hard fusion rules. It has been discussed that system performance in the global probabilities following the specific hard fusion rules is dependent on underlying parameter settings [19,22]. The drawback we need to compensate for is, on the basis of available ED results, to enhance the existing hard fusion rules policy for better fusion on local decisions, so that the system performance can be improved.

3.1 System Modeling

Assume that time is slotted and that the single channel in the considered flat-fading environment is unoccupied with the probability $Pr(\pi_0)$. In the traditional method, there's a set of local decisions LDs made by SUs, according to the energy of received signals over a period of time. Following the existing hard fusion rules, a final decision can be made to estimate PU's state. Without the ability of learning, the corresponding system performance cannot be adjusted, unless the underlying parameters of CRSN are changed. The correctness of each decision fusion jointly determines the system performance in global probability, which exactly matches the idea of dynamic programming, that is, the optimal solution is made up of and can be obtained by the optimal solutions to its sub-problems [24].

If the softly created Agent is trained for estimating PU's state, which targets selecting an appropriate final decision a from the set of available final decisions $FDs = \{1, 0\}$ at each time. As the bases for each fusion decision, the local decisions made by CRs/sensors can be organized into a matrix available for Agent's observation, recorded as state s. Obviously, the original traditional hard fusion rules are the policies that can be followed by the Agent.

Suppose local decisions made by SUs as the available state s, the value $q(s, a)$ of available final action $a \in FDs = \{1, 0\}$ made by the Agent in the recorded state s under a policy π can be expressed as

$$\begin{aligned} q_\pi(s, a) &\doteq \mathbb{E}_\pi \left[G_t \mid S_t = s, A_t = a \right] \\ &= \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a \right], \end{aligned} \tag{8}$$

where γ is the discount rate to determine the present value of future rewards. In addition, at the time t, R means the reward function based on state and action, and G_t represents the expected return, which can be further expressed by the end time T as

$$\begin{aligned} G_t &\doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-1} R_T \\ &= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \end{aligned} \tag{9}$$

Specifically, each action makes the same influence on the system performance, making it reasonable to set γ equal to 1. It's worth noting that the performance in global probability is obtained by statistics from a period of time, leading to the reward R being only the global probability after each preset period is achieved and being 0 for any state when the current period is incomplete.

When we seek to quantify the value of every fusion decision made during a long period of time, Monte Carlo reinforcement learning is a feasible solution to learn the state value from experienced completed state episodes by sampling, that is, estimating the real value of states with the average return. Theoretically, the more complete state episodes the Agent experiences, the more accurate the corresponding results will be [25].

3.2 Policy Evaluation and Improvement

With no existing convincing dataset for training, RL provides a feasible solution to achieve this goal by replacing FC in the CSS system as a softly-created Agent that can learn from past experiences. More than following originally fixed policies (traditional hard fusion rules), the Agent is capable of intelligently evaluating and improving policies so that global probabilities can be further increased. Due to the implementation of learning is based on the originally existing sensing and fusion process, no additional prior knowledge is still required during the spectrum sensing process.

For any policy π of the Agent, the process of calculating the state-value function v_π is called Policy Evaluation that can be expressed as

$$\begin{aligned} v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s]. \end{aligned} \tag{10}$$

As a prediction problem, the computation of the value function for a policy is core to assisting in the development of better policies. As we target to improve the performance on the basis of existing hard fusion rules, the iterative solution approach can be used to accomplish the updation of the policy as

$$v_{k+1}(s) \doteq \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s]. \tag{11}$$

Now that we have established the value function v_π for each given deterministic policy π, it's meaningful to learn whether the policy should be changed to choose an action $a \neq \pi(s)$ deterministically for some state s. A general approach is, for all $s \in S$, let π and π' be any pair of deterministic policies such that

$$\begin{aligned} \pi'(s) &\doteq \underset{a}{argmax}\ q_\pi(s,a) \\ &= \underset{a}{argmax}\ \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a], \end{aligned} \tag{12}$$

where the new greedy policy is taken into account and the action chosen at each state that $q_\pi(s,a)$ deems to be the best.

To ensure continuous 'Exploration' with greedy algorithms, a parameter ϵ was introduced to adjust the greedy rate so that all possible actions under a certain state have a non-zero probability of being selected and executed. Specifically, the action thought as the optimal one currently is selected with the probability of $(1-\epsilon)$, and the action chosen among all possible behaviors (including the current best behavior) with the probability of ϵ, which is expressed as

$$\pi(a \mid s) = \begin{cases} \epsilon/K + 1 - \epsilon \; if \; a^* = \underset{a \in A}{argmax}\ q(s,a) \\ \epsilon/K \qquad\qquad\qquad\qquad Others \end{cases} \tag{13}$$

where K is the number of available actions, which is 2 in this paper.

The CSS environment's dynamics determine the state transition probability is unknown, as well as both the action space and observation space are in discrete forms. The limited number of SUs in CRSN causes finite states that are observable for the Agent, that is, as long as sensing for enough times, all possible states will appear. After sampling one or more complete state episodes through the ϵ-greedy algorithms, Monte Carlo Control will average the value of the experienced State-Action pair, and continue to implement evaluation and improvement to the policy without a termination condition. More than avoiding losing any information and state, the policy is expected to continuously improve and finally arrive at optimal. There exists the theorem that Greedy in Limit with Infinite Exploration (GLIE) Monte Carlo Control can help a policy converge to the optimal state-action value function with increasing sampling, that is $q(s,a) \rightarrow q^*(s,a)$, with all state-action pairs being explored infinitely. As explained before, the policy iteration is accomplished through the value iteration, the specific process of the Monte Carlo Control based on GLIE can be expressed as follows. Specifically, for time $t \in [1,T]$, the reward r_{t+1} represents the instant reward received by the Agent when the Agent leaves the state s_t after taking the action a_t. As explained before, instant rewards only appear at the end of a full episode, so rewards in intermediate states are set to 0. In addition, the value of ϵ is gradually decreased with the increase of time as $\epsilon = \frac{1}{t}$, where t denotes the total times of learning the Q-value from the state sequence. Apart from such a way, an adaptive ϵ-greedy algorithm was also proposed to adjust the ϵ automatically. Considering that no significant gain when $\epsilon > 0.5$ [26], as detailed in Algorithm 2, l and f are two customizable parameters to alter the value of ϵ in the traditional ϵ-greedy exploration mode. Specifically, the number of exploration mode iterations to run before modifying ϵ can be controlled by l. The parameter f is in charge of regularizing the received calculated values of the rewards to ensure that the freshly created value of ϵ by the function is appropriate. The sigmoid function used to reset a new ϵ can be expressed by $sigmoid(x) = \frac{1.0}{1.0+exp(-2*x)} - 0.5$.

Algorithm 1. Monte Carlo Control based on GLIE

Sampling k-th state sequence $S_k = \{s_1, a_1, r_2, \cdots, s_T\}$ based on the given policy π
for all $(s_t, a_t) \in S_k$ **do**
  Update its explored times and value function as
    $n(s_t, a_t) \leftarrow n(s_t, a_t) + 1$
    $q(s_t, a_t) \leftarrow q(s_t, a_t) + \frac{1}{n(s_t, a_t)} (G_t - q(s_t, a_t))$
  Improve the policy based on the latest action value function as
    $\epsilon \leftarrow \frac{1}{k}$
    $\pi \leftarrow \epsilon - greedy(Q)$
end for

Algorithm 2. Adaptive ϵ-greedy

$max_{prev} \leftarrow 0$
$k \leftarrow 0$
if standard normal distribution $N(0,1) \leq \epsilon$ **then**
  $max_{curr} \leftarrow q^*(s_t, a_t)$
  $k \leftarrow k+1$
  if $k = l$ **then**
    $\Delta \leftarrow (max_{curr} - max_{prev}) * f$
    if $\Delta > 0$ **then**
      $\epsilon \leftarrow sigmoid(\Delta)$
    else
      if $\Delta < 0$ **then**
        $\epsilon \leftarrow 0.5$
      end if
    end if
    $max_{curr} \leftarrow max_{prev}$
    $k \leftarrow 0$
  end if
  randomly selects an action
else
  select A_t^*
end if

4 Performance Validation

4.1 Experimental Settings

The parameters to construct a CRSN environment are summarized in Table 1. Settings are almost consistent with [27], we additionally double the range of sensors' SNRs to be within 15 dB to 30 dB, so that experiments can be conducted under multiple environments for a more comprehensive comparison. Denote π_0 as the probability that PU is absent. We consider $\pi_0 = \{0.2, 0.5, 0.8\}$ to represent situations where PU exists at a respective high, middle, and low frequency. In addition, a probability varying from 0 to 1 is used to represent situations where PU exists at a random and unfixed frequency during the spectrum sensing process. The regularization parameter f and times of exploration mode before changing ϵ in the adaptive ϵ-greedy algorithm are consistent with [26].

We set the every 50 times sensing process as one complete episode so that the softly-created Agent can learn from it. During each complete episode, it's worth noting that the underlying parameters (including the SNR of each sensor $SNR_1, \cdots, SNR_M$, thresholds for ED λ_1, λ_2, and sleeping rates $\mu_1, \cdots, \mu_M$) refreshed at the beginning and kept unchanged in the sensing process. As the softly-created Agent estimates the PU's state based on the traditional hard fusion rules and keeps improving the policy for decision-making with sensing times, once the local decisions are made, the trained Agent can make fusion decisions according to the policy it masters.

Table 1. Detailed parameters for experimental environments to implement traditional methods and the Monte Carlo Control based on which with ϵ-greedy algorithms.

	Symbol	Description	Value
CRSN	$Pr(\pi_0)$	The probability of PU absent	0.2, 0.5, 0.8, changing
	N	Number of samples	5
	M	Number of sensor nodes	5
	γ_j	Signal-to-noise ratio of the j-th sensors	[0, 15], [15, 30] dB
	μ_j	Sleeping rate of the j-th sensor	(0, 1)
traditional methods	λ_1	Lower threshold in Energy Detection	[0, 30]
	λ_2	Upper threshold in Energy Detection	$[\lambda_1, 60]$
	k	Majority threshold in 'Majority' rule	0.5
MCRL	ϵ	Exploring rate in greedy algorithm	decreasing, adaptive
	l	Times of exploration mode before changing ϵ	10
	f	Regularization parameter for an adequate ϵ	7
	γ	Discount rate to determine each reward equals	1

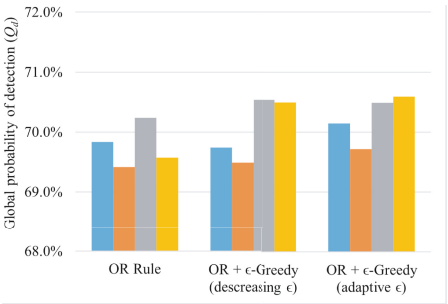
As explained before, the purpose of the proposed method is to help improve system performance in Q_d and Q_f by implementing Monte Carlo Control on existing traditional hard fusion rules. With traditional hard fusion rules, Q_d and Q_f can be calculated by the underlying parameters of the sensor through the formulas [19,22], which should also be consistent with the result achieved by the statistical way as described in Eq. (6) and Eq. (7), if PU's state in each time slot is known. During the training process, we direct the Agent's learning by letting the Agent go through 600 complete episodes, i.e., 30,000 times sensing so that it can experience all possible states in the simulated CSS environment where PU's states are known for further calculating Q_d and Q_f as the instant reward. At the same time, the original 3 hard fusion rules can be respectively improved with Monte Carlo Reinforcement Learning, so that the Agent knows when to follow the original rules or not. While during the testing process, the policy stops improving, that is, Q_d and Q_f at each complete episode are only recorded for visualizing the performance and don't guide the Agent to learn anymore.

4.2 Global Probability of Detection

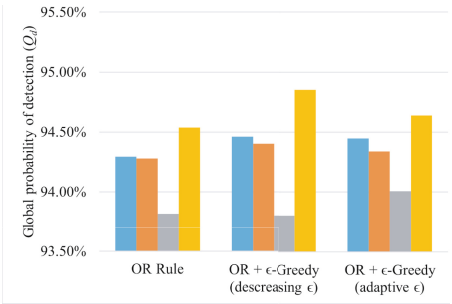
To validate the effectiveness of our method, with 3 fixed (including 0.2, 0.5, 0.8) and one changing $Pr(H_0)$, a total of 400 complete episodes containing 20,000 times sensing after improvement are recorded for statistics to compare with fixed traditional hard fusion rules, as shown in Fig. 2.

It's evident that with naive traditional methods, the performances in Q_d under higher SNRs are superior to those under lower SNRs, which verified again that the traditional methods' good or not indeed rely on the underlying parameters.

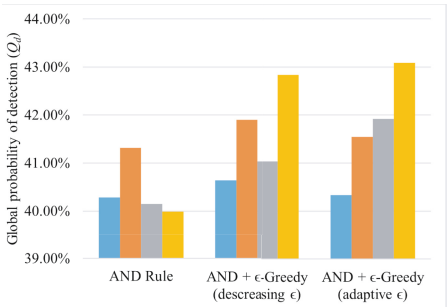
Under the 'OR' fusion rule, our proposed methods with various ϵ values more or less outperform the traditional one, when faced with higher SNRs. When SNRs decrease, ϵ-greedy algorithms with a decreasing ϵ may be unstable due to their



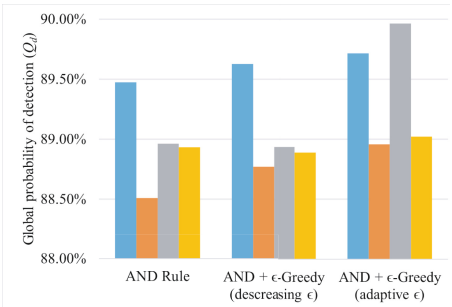
(A1) 'OR' Rule, $\gamma_j \in [0, 15]$



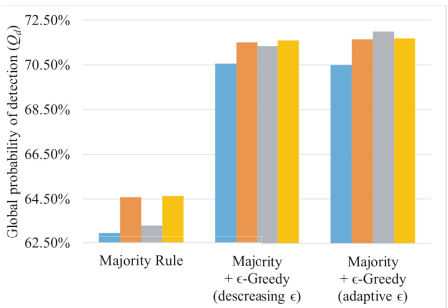
(B1) 'OR' Rule, $\gamma_j \in [15, 30]$



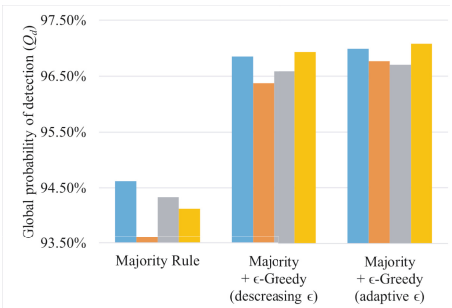
(A2) 'AND' Rule, $\gamma_j \in [0, 15]$



(B2) 'AND' Rule, $\gamma_j \in [15, 30]$



(A3) 'Majority' Rule, $\gamma_j \in [0, 15]$



(B3) 'Majority' Rule, $\gamma_j \in [15, 30]$

Fig. 2. The global probabilities of detection Q_d under different methods. (A1)–(A3) and (B1)–(B3) correspond to sensors under environments with low and high SNRs respectively.

incapability of adjusting the learning rate, while the adaptive ϵ always holds its advantage in learning to adjust Q_d and achieves the best performance.

If the 'AND' fusion rule is adopted, with lower SNRs, both 2 ϵ-greedy algorithms always help improve the Q_d, especially with a changing $Pr\,(H_0)$. Under

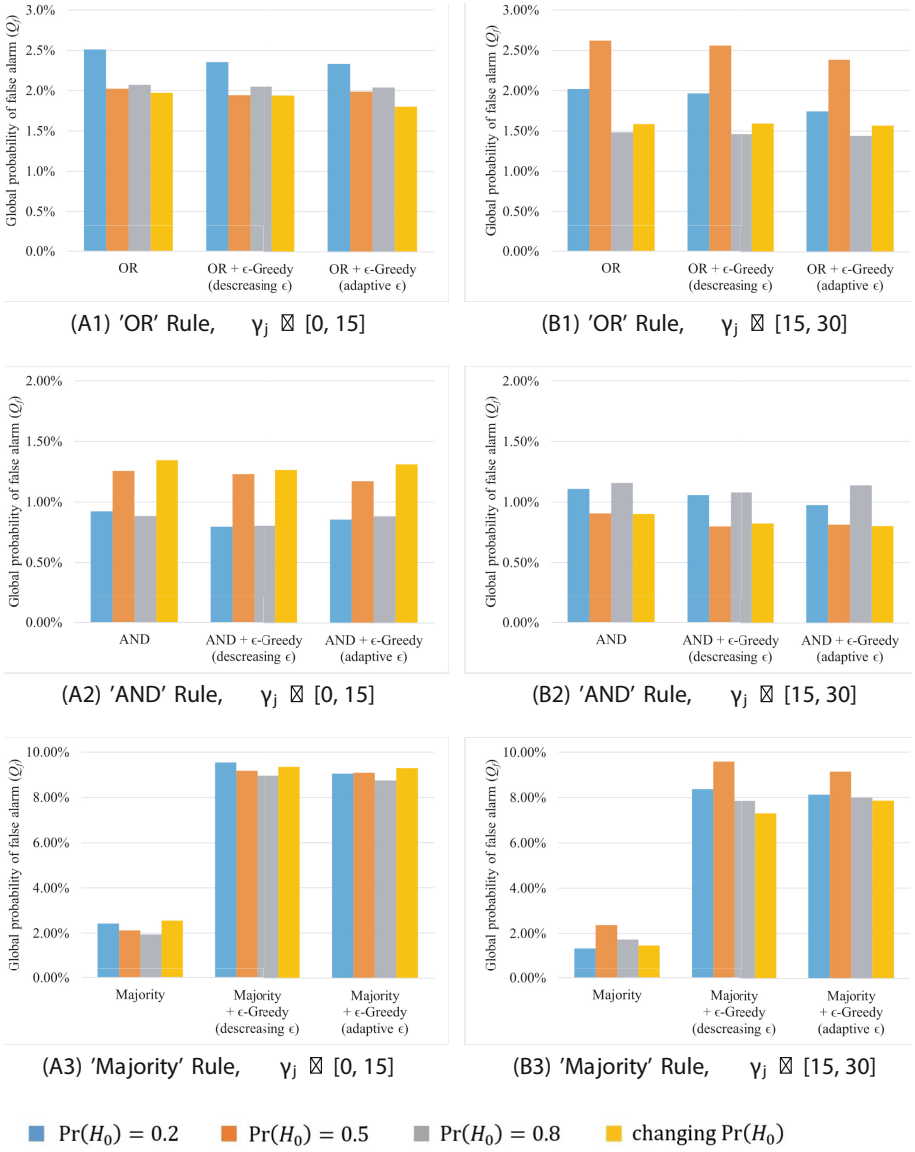


Fig. 3. The global probabilities of false alarm Q_f under different methods. (A1)–(A3) and (B1)–(B3) correspond to sensors under environments with low and high SNRs respectively.

higher sensors' SNRs, the combination of the naive fusion rule and the adaptive ϵ-greedy algorithm always seems to be the best with all 4 different $Pr\,(H_0)$, from which the still existing instabilities of the greedy algorithms with a non-adaptive

ϵ can be found. Due to its inherent property, the Q_d using the 'AND' rule is much lower than that using 'OR' rule.

Based on the 'Majority' fusion rule, all 2 ϵ-greedy algorithms improve the Q_d a lot, which can be up to approximately 9% with lower SNRs. In addition, the greedy algorithm with an adaptive ϵ value achieves the best performance in most situations.

4.3 Global Probability of False Alarm

Apart from Q_d, Q_f is another metric that can reflect the CSS performance. In RL, our goal is always chasing the reward maximization. As mentioned before, the reward is consistent with Q_d when we test the proposed method's effectiveness in improving Q_d, both of them are expected to be maximized. Similarly, when the proposed method is applied for chasing lower Q_f, we set the reward to be consistent with the negative number of Q_f after each fusion decision. In this way, the smaller the Q_f is, the bigger the corresponding reward can be prized. The same tests are implemented to estimate the decrease in Q_f, the averaged Q_f of 400 episodes tests with various fusion rules are depicted in Fig. 3.

Under the 'OR' and 'AND' fusion rules, after introducing ϵ-greedy algorithms to adjust the traditional hard fusion rules, there is a slight drop or remains unchanged in Q_f most of the time.

When it comes to the 'Majority' rule, the utilization of the ϵ-greedy algorithms increased the Q_f. Backtracking the system performance in Q_d using the 'Majority' Rule, we can find that the improvement based on this traditional hard fusion rule is the most obvious and effective. As explained before in Eq. (6) and Eq. (7), both the Q_d and Q_f are dependent on the probability of the situation that the fusion decision declaring PU present. The calculation method determines the trade-off that exists between the two system performance metrics. Comparing the results among ϵ-greedy algorithms with different ϵ, the one in an adaptive way still outperforms others under most environments.

5 Conclusion and Future Work

In this paper, the heterogeneous features of PU and SU are investigated. With the observation of local decisions made by SUs, the traditional hard fusion rules were improved by the Monte Carlo Control with the goal of improving the CSS performance. Specifically, the traditional FC without the ability of learning was replaced with a softly-created Agent, which can take ϵ-greedy policies during making fusion decisions. The proposed method can be considered to complement existing methods and apply them to satisfy different requirements in various scenarios. However, since the MCRL in this paper is implemented based on local decisions made by SUs that may exist errors, leading to still room for improvement in CSS performance. Although the CRSN network topology and routing are assumed to be static in this paper, our methodology can potentially be used for other CRSN topologies and schemes, which will be a subject of future work.

References

1. Lin, J., Wei, Yu., Zhang, N., Yang, X., Zhang, H., Zhao, W.: A survey on internet of things: architecture, enabling technologies, security and privacy, and applications. IEEE Internet Things J. **4**(5), 1125–1142 (2017)
2. Fang, K., Wang, T., Zhou, X., Ren, Y., Guo, H., Li, J.: A topsis-based relocalization algorithm in wireless sensor networks. IEEE Trans. Industr. Inf. **18**(2), 1322–1332 (2021)
3. Lien, S.-Y., Cheng, S.-M., Shih, S.-Y., Chen, K.-C.: Radio resource management for QoS guarantees in cyber-physical systems. IEEE Trans. Parallel Distrib. Syst. **23**(9), 1752–1761 (2012)
4. Kolodzy, P.: Spectrum policy task force. Federal Communications Commission, Washington, DC, Report ET Docket, vol. 40, no. 4, pp. 147–158 (2002)
5. Cichoń, K., Kliks, A., Bogucka, H.: Energy-efficient cooperative spectrum sensing: a survey. IEEE Commun. Surv. Tutor. **18**(3), 1861–1886 (2016)
6. Urkowitz, H.: Energy detection of unknown deterministic signals. Proc. IEEE **55**(4), 523–531 (1967)
7. Chaudhari, S., Lunden, J., Koivunen, V., Poor, H.V.: Cooperative sensing with imperfect reporting channels: hard decisions or soft decisions? IEEE Trans. Signal Process. **60**(1), 18–28 (2011)
8. Sun, C., Zhang, W., Letaief, K.B.: Cluster-based cooperative spectrum sensing in cognitive radio systems. In: 2007 IEEE International Conference on Communications, pp. 2511–2515. IEEE (2007)
9. Armi, N., Saad, N.M., Arshad, M.: Hard decision fusion based cooperative spectrum sensing in cognitive radio system. ITB J. Inf. Commun. Technol. **3**(2), 109–122 (2009)
10. Nallagonda, S., Kumar, Y.R., Shilpa, P.: Analysis of hard-decision and soft-data fusion schemes for cooperative spectrum sensing in rayleigh fading channel. In: 2017 IEEE 7th International Advance Computing Conference (IACC), pp. 220–225. IEEE (2017)
11. Varshney, P.K.: Distributed Detection and Data Fusion. Springer, New York (2012)
12. Fang, K., Yang, B., Zhu, H., Lin, Z., Wang, Z.: Two-way reliable forwarding strategy of RIS symbiotic communications for vehicular named data networks. IEEE Internet Things J. **10**(22), 19385–19398 (2022)
13. Zhu, H., Peng, Y., Xu, H., Tong, F., Jiang, X.-Q., Mirza, M.M.: Secrecy enhancement for SSK-based communications in wireless sensing systems. IEEE Sens. J. **22**(18), 18192–18201 (2022)
14. Ng, A.Y.: Shaping and policy search in reinforcement learning. University of California, Berkeley (2003)
15. He, H., Jiang, H.: Deep learning based energy efficiency optimization for distributed cooperative spectrum sensing. IEEE Wirel. Commun. **26**(3), 32–39 (2019)
16. Zhu, J., Song, Y., Jiang, D., Song, H.: Multi-armed bandit channel access scheme with cognitive radio technology in wireless sensor networks for the internet of things. IEEE Access **4**, 4609–4617 (2016)
17. Lelarge, M., Proutiere, A., Talebi, M.S.: Spectrum bandit optimization. In: 2013 IEEE Information Theory Workshop (ITW), pp. 1–5. IEEE (2013)
18. Tehrani, P., Zhao, Q., Tong, L.: Multi-channel opportunistic spectrum access in unslotted primary systems with unknown models. In: 2011 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), pp. 157–160. IEEE (2011)

19. Maleki, S., Leus, G., Chatzinotas, S., Ottersten, B.: To and or to or: how shall the fusion center rule in energy-constrained cognitive radio networks? In: 2014 IEEE International Conference on Communications (ICC), pp. 1632–1637. IEEE (2014)
20. Lu, L., Zhou, X., Onunkwo, U., Li, G.Y.: Ten years of research in spectrum sensing and sharing in cognitive radio. EURASIP J. Wirel. Commun. Networking **2012**(1), 1–16 (2012)
21. Maleki, S., Pandharipande, A., Leus, G.: Energy-efficient distributed spectrum sensing with convex optimization. In: 2009 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), pp. 396–399. IEEE (2009)
22. Maleki, S., Leus, G., Chatzinotas, S., Ottersten, B.: To and or to or: on energy-efficient distributed spectrum sensing with combined censoring and sleeping. IEEE Trans. Wireless Commun. **14**(8), 4508–4521 (2015)
23. Peh, E.C.Y., Liang, Y.-C., Guan, Y.L., Pei, Y.: Energy-efficient cooperative spectrum sensing in cognitive radio networks. In: 2011 IEEE Global Telecommunications Conference-GLOBECOM 2011, pp. 1–5. IEEE (2011)
24. Vince, A.: A framework for the greedy algorithm. Discret. Appl. Math. **121**(1–3), 247–260 (2002)
25. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: a survey. J. Artif. Intell. Res. **4**, 237–285 (1996)
26. dos Santos Mignon, A., de Azevedo da Rocha, R.L.: An adaptive implementation of ε-greedy in reinforcement learning. Procedia Comput. Sci. **109**, 1146–1151 (2017)
27. Wu, Q., Ng, B.K., Lam, C.-T.: Energy-efficient cooperative spectrum sensing using machine learning algorithm. Sensors **22**(21), 8230 (2022)