Reinforcement Learning-Based Cooperative Spectrum Sensing

Wenli Ning(✉), Xiaoyan Huang, Fan Wu, Supeng Leng,
and Lixiang Ma

School of Information and Communication Engineering,
University of Electronic Science and Technology of China,
Chengdu 611731, China
WenliNing@126.com,
{xyhuang, wufan, spleng, lixiangma}@uestc.edu.cn

Abstract. In cognitive radio (CR) networks, the detection result of a single user is susceptible due to shadowing and multipath fading. In order to find an idle channel, the secondary user (SU) should detect channels in sequence, while the sequential detection may cause excessive overhead and access delay. In this paper, a reinforcement learning (RL) based cooperative sensing scheme is proposed to help SU determine the detection order of channels and select the cooperative sensing partner, so as to reduce the overhead and access delay as well improve the detection efficiency in spectrum sensing. By applying Q-Learning, each SU forms a dynamic priority list of the channels based on neighbors' sensing results and recent act-observation. When a call arrives at a SU, the SU scans the channel in list order. To improve the detection efficiency, the SU can select a neighbor with potential highest detection probability as cooperative partner using multi-armed bandit (MAB) algorithm. Simulation results show that the proposed scheme can significantly reduce the scanning overhead and access delay, and improve the detection efficiency.

Keywords: Spectrum sensing · Reinforcement learning · Cooperative sensing · Q-Learning · Multi-armed bandit

1 Introduction

In wireless networks, inefficient and fixed spectrum usage mode is the main reason for low utilization of spectrum resources. CR technology is envisaged to solve this problem by exploiting the existing wireless spectrum opportunistically [1, 2]. In CR networks, SU can opportunistically transmit in the vacant portions of the spectrum already assigned to licensed primary users (PUs). The goal of spectrum sensing is to find idle spectrum for SUs to occupy while reducing the interference to PUs.

There are two main problems in spectrum sensing. Firstly, due to the detection errors caused by fading and shadowing, the local detection result of a single user on a channel is susceptible [3]. Secondly, we usually use energy detection in local detection. But when there is a demand, SU needs to detect the licensed channels in sequence until it finds an available channel, which can cause excessive overhead and access delay.

B. Li et al. (Eds.): IoTaaS 2018, LNICST 271, pp. 149–161, 2019.
https://doi.org/10.1007/978-3-030-14657-3_16

Hence, selecting the most likely idle channel to sense can reduce the scanning overhead and access delay.

The problems above cause serious access delay, overhead and inefficiency in spectrum sensing. Cooperative spectrum sensing technology [4, 5] has been used in CR network to improve the detection efficiency. Authors in [6] proposed that when a SU's detection ability is higher than the other nodes, taking the local decision of this node as the final decision can obtain better performance than cooperation. So in this case, SU hopes the neighbor with highest detection ability can help him detect the channel. Reinforcement Learning [7] techniques are often applied in dynamic environment to maximum rewards, Q-Learning [8] and multi-armed bandit [9, 10] are two of the RL algorithms. In order to alleviate scanning overhead and access delay in spectrum access, authors in [11] use Q-Learning technique to estimate channels states based on the past history of channel usage. In [12], authors use Q-Learning to select independent users under correlated shadowing for cooperation to improve detection efficiency. In [13], authors formulate the online sequential channel sensing and accessing problem as a sequencing multi-armed bandit problem to improve the throughput.

To address the issues of access delay, scanning overhead and inefficiency in spectrum sensing, a novel cooperative sensing scheme based on RL is designed in this paper. Reinforcement learning is an online learning algorithm. The action-taking agent interacts with the external environment through reward mechanisms, and then adjusts its action according to the reward values. The aim of the agent is to learn the optimal action to maximize the reward. In our scheme, each SU is an agent who needs to learn the behaviors of channels and neighbors, and then takes action to improve the spectrum sensing performance.

Our contributions can be summarized as follows:

- We propose a channel status prediction algorithm based on Q-Learning for SUs to determine the detection order of channels. Specifically, each SU learns the channel patterns by neighbors and detection results. A dynamic priority list of the channels is formed accordingly during the learning procedure. Whenever there is a demand, an SU probes the channels in list order.
- We propose a cooperative partner selection algorithm based on MAB for SUs. Each SU estimates the detection probabilities of its neighbors by MAB algorithm. When detecting, the SU can select a neighbor with potential highest detection probability to help it sense the spectrum.
- Simulation results show that the proposed RL-based cooperative sensing scheme can greatly improve the performance in terms of the access delay, scanning overhead, and detection efficiency.

The remainder of this paper is organized as follows: Sect. 2 describes the system model. Section 3 elaborates the proposed RL-based cooperative sensing scheme. Section 4 evaluates the performance of the proposed scheme. Finally Sect. 5 concludes the paper.

2 System Model

We consider a CR network as shown in Fig. 1. We assume there are N SUs randomly distributed in the network. Each SU can communicate control packets with its neighbors over a channel of the ISM band, which is known to every node. PU network has L licensed channels. PUs may appear in a set of licensed channels. Due to the random distribution of SUs, the effects of fading and shadowing between each SU and PU are different. Thus the detection probability of each node is different.

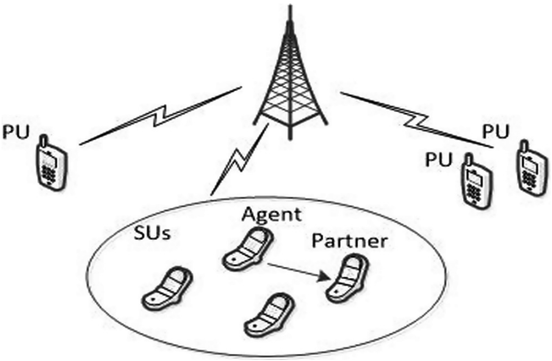

Fig. 1. CR network

To improve channel utilization, SUs attempt to find available spectrum to access by spectrum sensing. When there is a demand, SU needs to scan the licensed channels in sequence until it finds an available channel. In order to find the idle channel quickly, SUs use Q-Learning technique to predict the availability of a channel in our scheme. Q-value in Q-Learning technique represents the probability of each channel being idle. A dynamic priority list of channels is formed according to Q-values of all channels. When there is a demand at a SU, the SU takes an action by scanning a channel in the order of priority list, and then calculates the reward based on neighbors' sensing results and local detection. Then the SU uses the reward to update Q-value of this channel. Finally, the priority list is updated based on the updated Q-value. Whenever there is an update of channel status, the SU shares it with its neighbors.

However, because the multipath fading and shadowing effects in wireless channels can lead to detection errors, the sensing result obtained by a single SU is susceptible. Cooperative spectrum sensing can effectively combat shadowing and multipath fading. When a node cooperates with the partner with lower detection probability, the partner likely degrades the detection performance [7]. So in our scheme, SU would select the neighbor with highest detection ability as its cooperative partner. The selected cooperative neighbor will perform local energy detection, and then send its local binary decisions 1/0 to the SU. 1 and 0 indicate the absence and presence of the PU on the detected channel respectively.

3 RL-Based Cooperative Sensing Scheme

3.1 Q-Learning Based Channel Status Prediction Algorithm

When SU needs access to the channel, the SU hopes to choose the most likely idle channel to detect. Q-learning technique is applied to predict the statuses of channels and form a priority list of channels accordingly. When there is a demand, the SU can detect the channel using action select strategy according to the list, so as to reduce access delay and scanning overhead. Specially, when detecting, each SU select a channel according to the list, and then computes reward of the channel based on neighbors' sensing results and local detection. Then, SU uses reward to update Q-value of the channel. Q-value represents the estimated probability of the channel being idle. The dynamic priority list is updated according to Q-values of all channels.

Q-Learning. Q-Learning is a RL algorithm that includes two entities: agent and environment. An agent in a state s interacts with the environment by taking an action $a \in A$, and then the agent receives a reward $r(s, a)$. So the agent uses $r(s, a)$ to update $Q(s, a)$ and goes in state s'. The agent learns from the state-action-reward history. $Q(s, a)$ is updated in every iteration using the following formula:

$$Q(s, a) = (1 - \alpha)Q(s, a) + \alpha\left\{ r(s, a) + \beta \max_{b \in A}[Q(s', b)] \right\} \tag{1}$$

Here a is the learning rate, $0 \leq \alpha \leq 1$. With α closer to 0, the agent learns less from instant rewards and concentrates more on the history. β is the discount factor, $0 \leq \beta \leq 1$, which notes the attenuation of rewards in the future.

In this paper, agent is each SU. SU_k represents the secondary user k, $1 \leq k \leq N$. State indicates the occupancy statuses of all channels in the primary network. When a channel status turns to busy from idle or turns to idle from busy, the state changes. So the state changes dynamically as the PUs occupy the channel or not. An action is a decision an agent makes in a state. That SU_k chooses an action $a = c_i$, $1 \leq i \leq L$ indicates SU_k selects the c_i as the channel to be detected. How to choose actions depends on the action selection strategy. The choice of current action is evaluated by the reward. Reward function $r(s, a)$ maps the state-action transition to a real-valued reward. Considering the uncertainty of the result detected by a single user, thus SU_k calculates $r(s, a)$ based on both neighbors' sensing results and local detection.

Action Selection Strategy. When a call arrives at SU_k, SU_k selects c_i as the channel to be detected using ε-greedy strategy. That is, the channel with the highest priority according to the priority list will be selected with the probability of $1 - \varepsilon$, which is called exploitation. The channel will be randomly selected with the probability of ε, which is called exploration. ε controls the degree of exploration versus exploitation. For large ε, SU_k concentrates more on exploring the statuses of more channels, so as to help find potentially idle channels. For small ε, SU_k concentrates more on exploiting the current knowledge to perform current best selections, so as to reduce scanning overhead. The ε - greedy strategy helps SUs adapt to the channels' dynamic statuses.

The ε - greedy strategy is an improvement of the algorithms in [11], which helps SUs adapt to the dynamic statuses of channels to choose an idle channel.

Reward Calculation. The reward is used for evaluating the choice of current action. Suppose at time t, SU_k chooses the channel c_i according to the action selection strategy, and then performs local detection on the selected channel c_i. If the detection result is idle, SU_k attempts to access the channel and then obtains the access result. Considering the uncertainty of the result detected by a single user, SU_k calculates the $r^k(s_t, c_i)$ combining local result and detection results of its neighbors:

$$r^k(s_t, c_i) = \begin{cases} 1 - \sum\limits_{j=1}^{N} \frac{(1-s^j(c_i)) * W_t^j(c_i)}{N}, & \textit{if} \;\; s^j(c_i) = 1 \\ -\sum\limits_{j=1}^{N} \frac{(1-s^j(c_i)) * W_t^j(c_i)}{N}, & \textit{if} \;\; s^j(c_i) = 0 \end{cases} \tag{2}$$

Here $s^j(c_i)$ is the sensing result of c_i obtained by SU_j, if the detection result is idle and SU_k accesses c_i successfully, $s^j(c_i) = 1$, otherwise $s^j(c_i) = 0$. $W_t^j(c_i)$ is the detection weight of SU_j about c_i at time t, it represents the estimated value of SU_j's detection probability, which would be obtained by cooperative partner selection algorithm. We'll discuss it in the next section.

After computing the reward point, SU_k updates the corresponding Q-value of channel c_i as:

$$Q^k(s_{t+1}, c_i) = (1-\alpha) \cdot Q^k(s_t, c_i) + \alpha \cdot \left\{ r^k(s_t, c_i) - \beta(exp^{-\tau \cdot m}) \right\} \tag{3}$$

Here α is the learning rate, $0 \leq \alpha \leq 1$. β is the discount factor, $0 \leq \beta \leq 1$. τ, $0 \leq \tau \leq 1$ is a constant. m represents that it's the mth attempt for SU_k to find an idle channel.

The Q-learning based channel status prediction algorithm is applied for SUs to determine the detection order of channels. Each node maintains a Q table which consists of the Q-values of all channels. Q table is initialized to zero at start. When a call arrives at SU_k, SU_k selects a channel by ε-greedy strategy according to the priority list. Then SU_k detects the selected channel and attempts to access it if the detection result is idle. After that, SU_k calculates the reward and uses it to update Q-value of this channel. Based on the updated Q-value, SU_k prepares the new dynamic priority list of the channels for the next round. Cycling until SU_k accesses an idle channel or reaches the maximum number of attempts. The updated channel's status and corresponding weight are broadcasted in each round.

3.2 Multi-armed Bandit Based Cooperative Partner Selection Algorithm

In this paper, we assume that the detection probability of each SU is fixed. In order to increase the detection efficiency, when a call arrives at SU_k, SU_k hopes select a neighbor with highest detection probability to help it sense the spectrum. Since the detection probabilities of SUs are unknown, MAB technique is applied for SUs to estimates the detection probabilities of its neighbors. It also can enable users to learn a strategy to select cooperative partner to maximize detection probability.

MAB. The MAB problem is the model of a gambler (agent) who is playing a slot machine (arm). At time t, the agent gets a reward R_t by pulling/selecting arm $a \in A$. R_t has an independent and appropriate distribution. MAB has two stage of exploration and exploitation, the agent collects information of arms in exploration stage and exploits it in exploitation stage. The purpose of agent is to maximize the total rewards during pulling the arms.

In this paper, the agent is each SU. Arms are its neighbors which are the potential partners of the SU. Let f represent the arm, $1 \leq f \leq N$. $a = f$ means SU selects SU_f as its cooperative partner. Reward R_t represents the detection result of SU_f is right or not, $R_t = 1$ means SU_f detects correctly, or $R_t = 0$ means that SU_f makes a wrong conclusion of the channel status. The expected reward $p(f)$ represents the detection probability of SU_f, $p(f) = E[R|a = f]$, we call it the true value. $\hat{p}_t(f)$ represents the estimation of $p(f)$ at time t. $\hat{p}_t(f)$ is calculated by the obtained information, we call it the estimated value.

MAB has two stage of exploration and exploitation. In exploration stage, agent can obtain more information of arms for selecting better arms, in exploitation stage agent can use the obtained information to maximize its current reward. But when the algorithm focuses more on exploitation, it produces regrets. When the algorithm focuses more on exploitation, it can't find better arms. This is the exploration versus exploitation dilemma. Bandit algorithms look for a balance between exploration and exploitation.

Sample Mean Method. Because the detection probability of SU_f doesn't change with time in our scheme, so it's reasonable to choose the mean of the samples as $\hat{p}_t(f)$, $\hat{p}_t(f)$ is calculated by formula (4):

$$\hat{p}_t(f) = \frac{1}{t}\sum_{i=1}^{t} R_i \tag{4}$$

Action Selection Strategy. Upper confidence bound (UCB) algorithm takes into account both estimated value and selection times of each action to explore and exploit. The aim of UCB algorithm is to choose the most potential user to achieve a balance between exploration and exploitation. At each time t, the action is selected by following formula:

$$a_t = arg_f max\left[\hat{p}_t(f) + c\sqrt{\frac{logt}{N_t^f}}\right] \tag{5}$$

Here c controls the degree of exploration versus exploitation. N_t^f represents the times that SU_f has been selected as partner until time t, if $N_t^f = 0$, SU_f will be chosen firstly.

The MAB based algorithm is applied for each SU to select a neighbor for cooperation. If there is a demand at SU_k. Firstly, SU_k selects a cooperative partner SU_f using formula (5). And then, SU_k selects a channel c_i using Q-learning algorithm. SU_f performs local energy detection and sends its local binary decisions 1/0 to SU_k. SU_k attempts to access c_i if SU_f's decisions is 1. According to the detection and access result, SU_k updates the estimated value $\hat{p}_t(f)$ of SU_f. The updated $\hat{p}_t(f)$ is also the detection weight $W_t^k(c_i)$ of SU_k in channel status prediction algorithm. Then, SU_k continues to performs Q-Learning based algorithm.

3.3 RL-Based Cooperative Sensing Scheme

In summary, the proposed RL-based cooperative sensing scheme consists of the aforementioned Q-learning based channel status prediction algorithm and MAB based cooperative partner selection algorithm, as presented in Table 1.

Each node maintains a Q table which consists of the Q-values of all channels. Q table is initialized to zero at start. When a call arrives at a node SU_k, the main flow of the proposed scheme is as follows:

(1) SU_k selects a channel c_i using ε-greedy according the priority list, and then selects a cooperator SU_f using UCB.

(2) Update the $\hat{p}_t(f)$ or $W_t^k(c_i)$ by MAB algorithm according to the detection result.

(3) Update the priority list of channels by Q-Learning algorithm according to the sensing result.

(4) Broadcast the updated channel's status, the corresponding weights $W_t^k(c_i)$ and $\hat{p}_t(f)$ to its neighbors. Loop 1–3 until SU_k accesses an idle channel or reaches the maximum number of attempts.

The time complexity of channel selection strategy and cooperative partner selection strategy are $O(L)$ and $O(M)$ respectively. Here, M is the maximum number of attempts for each call before declaring a call block. L is the total number of channels in primary network. It can be seen from Algorithm 1, if we consider the worst case, the time complexity of each SU for one call is $O(M(L+N))$. N is the total number of neighbors of one SU.

Table 1. Pseudo code of the proposed algorithm.

```
Algorithm 1: RL-based Cooperative Sensing Scheme

Input: the set of SUs, $W_{t-1}^{k}(c_i)$ and $Q^{k}(s_{t-1},c_i)$ of each $SU_k$
for all $c_i$, $\hat{p}_{t\text{-}1}(f)$ of each $SU_k$ .
Output: $s^k(c_i)$, $W_t^k(c_i)$, $\hat{p}_t(f)$
for each $SU_k$ do
  if (a demand appears) then
      success =0; attempt=0;
      repeat
          Select a channel $c_i$ using $\varepsilon$-greedy;
          Select a cooperator $SU_f$ using UCB;
          if ($SU_f$ detects $c_i$ correctly) then
              $R_t = 1$;
          else
              $R_t = 0$;
          end
          Update $\hat{p}_t(f)$;
          $W_t^k(c_i) = \hat{p}_t(f)$;
          if ($SU_k$ access $c_i$ successfully) then
              $s^k(c_i) = 1$
              $r^k(s_t,c_i) = 1 - \sum_{j=1}^{N} (1 - s^j(c_i)) * W_{t-1}^{j}(c_i) / N$;
              success = 1;
          else
              $s^k(c_i) = 0$
              $r^k(s_t,c_i) = -\sum_{j=1}^{N} (1 - s^j(c_i)) * W_{t-1}^{j}(c_i) / N$;
          end
          Update $Q^k(s_t,c_i)$;
          ++attempt;
      until success =1 || attempt=M;
      if(success = 0)
          Declare call dropped.
      end
      Broadcast $s^k(c_i)$, $W_t^k(c_i)$ and $\hat{p}_t(f)$;
  end
end
```

4 Performance Evaluation

4.1 Simulation Setup

In this section, we evaluate the performance of the proposed scheme. In this paper, it is assumed that time is discrete with fixed time unit. In CR network, each SU has 4 neighbors and inquires whether there is a demand at each time unit. The arrival of call request follows Poisson process with $\lambda = 0.5/\mathit{time\ unit}$. There are 10 potential available channels, and PUs' usage rate of channels varies from 40% to 90% [11]. It is assumed that the maximum number of attempts of one call for each SU is 5, if the SU fails to access a channel for 5 times, the call is abandoned and announced blocked.

4.2 Effect of System Parameters

The parameter in the proposed scheme needs to be set according to the specific situations. *c* is the control parameter of MAB algorithm, which controls the degree of exploration versus exploitation. If *c* is too large or too small, the probability estimation of neighbors will be inaccurate, which will lead to inefficient cooperation. We can use the average detection probability to evaluate *c* of different values.

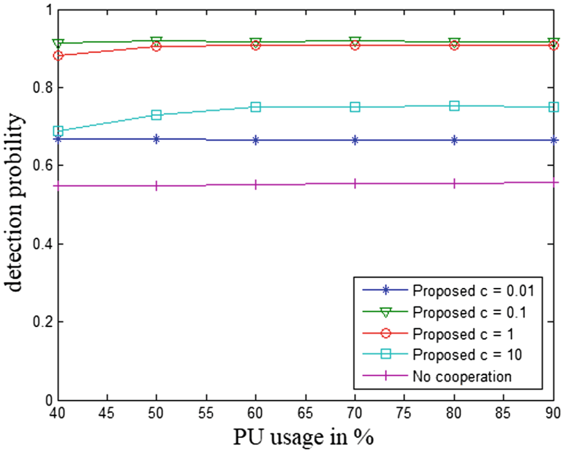

Fig. 2. Average detection probability versus PU usage for different parameter

Figure 2 shows the average detection probability versus PU usage. It can be seen from Fig. 2, cooperation can significantly improve the detection probability of SUs. The exploration coefficient *c* of MAB has a great influence on partner selection. When the value of *c* is set about 0.1, the algorithm achieves the balance of exploration and exploitation, so SU can select partner with high detection probability to cooperate. It reflects that if the exploration coefficient *c* is set properly, SU in our proposed scheme can indeed select partner with high detection probability.

4.3 Comparison with Other Algorithms

To evaluate the performance of the proposed scheme, we considered other two algorithms. The algorithm proposed in [11] (denoted as QLNC) estimates the status of channels based on the Q-Learning. The other algorithm uses Q-Learning approach to estimate the status of channels and then uses K/N rule to cooperate (QLKN). Figures 3, 4 and 5 compare the performance of our proposed scheme with other two algorithms.

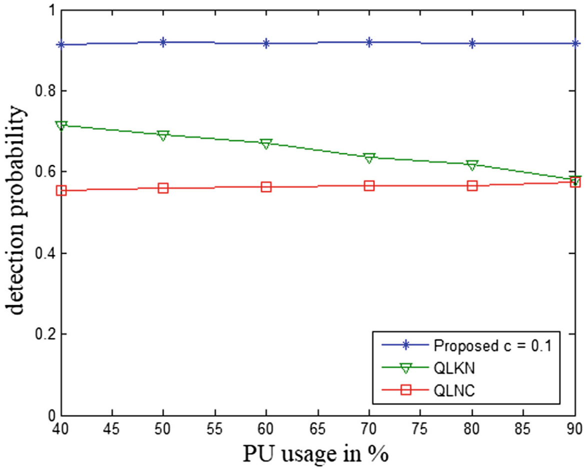

Fig. 3. Average detection probability versus PU usage for different algorithms

Figure 3 shows the average detection probability versus PU usage. It can be seen from Fig. 3, our proposed scheme performs much better than other two algorithms. This is because that when the parameter *c* is set to 0.1 in our simulation scene, the D-UCB algorithm can learn the dynamic detection probabilities of its neighbors well, thus SU can select the potential best neighbor to cooperate to improve the detection efficiency. So when the discount factor *c* is set properly in a specific dynamic situation, our proposed scheme can significantly improve the detection efficiency.

Figure 4 shows the average number of attempts for a successful access versus PU usage. It can be seen from Fig. 4, our proposed algorithm has the least average attempts in all the cases, and the average attempts increase with the PU usage in all the algorithms. This stems from the fact that Q-Learning technique forms a priory list of channels according to their statues, thus SU in our scheme just needs fewer times of detection to find an idle channel. With PU usage increasing, there are less opportunities for SUs to explore available channels in Q-Leaning based algorithm. So that the priority list can't be updated accurately. Average attempts reflect the scanning overhead and access delay, hence our proposed scheme indeed improves the scanning overhead and access delay.

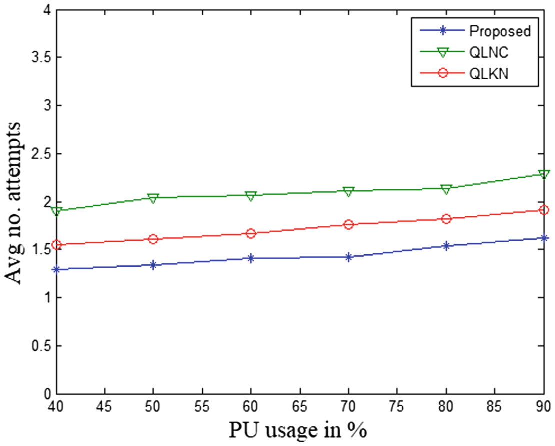

Fig. 4. Average attempts versus PU usage for different algorithms

Figure 5 shows average block rate versus PU usage. It can be seen from Fig. 5 that our proposed algorithm has the least average block rate in all the cases, the average block rate increases with the PU usage in all the algorithms. This stems from the fact that MAB algorithm can help SU learn the detection probabilities of its neighbors, thus SU can select the potential best neighbor to cooperate to improve the detection efficiency. With PU usage increasing, the decrease of the number of available channels leads to more exploration errors. Block rate reflects the quality of service provided to users, hence our proposed scheme performs better than the other two algorithms in terms of communication quality.

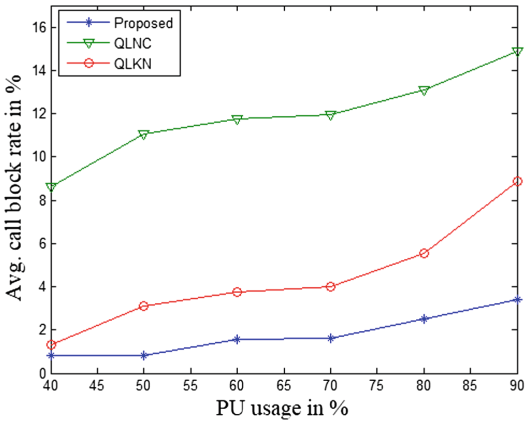

Fig. 5. Average block rates versus PU usage

4.4 Advantages and Disadvantages of the Proposed Scheme

According to the above simulation and analysis, the main advantages can be summarized as follows: the Q-Learning based channel status prediction algorithm can help SUs form a priority list of channels. When there is a demand, the SU can scan the channels in the list order, which helps reduce scanning overhead and access delay. MAB based cooperative partner selection algorithm can help SUs select a partner with high detection probability to cooperative. It improves the average detection probability. The proposed scheme also has some disadvantages: when we apply this scheme to a specific scenario, it takes some time to adjust the parameters. Also the scheme has a poor performance in scenes where the detection probabilities of SUs change dynamically.

5 Conclusion

In this paper, we proposed a RL based cooperative sensing scheme including the Q-Learning based channel status prediction algorithm and the MAB based cooperative partner selection algorithm. The Q-Learning based channel status prediction algorithm is applied for SUs to determine the detection order of channels. MAB based cooperative partner selection algorithm can help SUs select the neighbor with potential highest detection probability to cooperate. Simulation results demonstrate that compared to the existing algorithms (e.g., QLNC in [11] and QLKN), the proposed RL-based scheme has less scanning overhead, less access delay, and higher detection efficiency. In the future, effective learning strategies for mobile SUs will be studied.

Acknowledgement. This work is supported by the National Natural Science Foundation of China under Grant No. 61601083, and the Fundamental Research Funds for the Central Universities, China, No. ZYGX2016J001.

References

1. Wang, B., Liu, K.J.R.: Advances in cognitive radio networks: a survey. IEEE J. Sel. Topics Sig. Process. **5**(1), 5–23 (2011)
2. Haykin, S., Thomson, D.J., Reed, J.H.: Spectrum sensing for cognitive radio. In: Proceedings of the IEEE, pp. 849–877. IEEE (2009)
3. Uchiyama, H., Umebayashi, K., Kamiya, Y.: Study on cooperative sensing in cognitive radio based AD-HOC network. In: IEEE, International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1–5. IEEE, Athens (2007)
4. Akyildiz, I.F., Lo, B.F., Balakrishnan, R.: Cooperative spectrum sensing in cognitive radio networks: a survey. Phy. Commun. **4**(1), 40–62 (2011)
5. Mishra, S.M., Sahai, A., Brodersen, R.W.: Cooperative sensing among cognitive radios. In: IEEE International Conference on Communications, pp. 1658–1663. IEEE, Istanbul (2006)
6. Zheng, Y., Xie, X., Yang, L.: Cooperative spectrum sensing based on SNR comparison in fusion center for cognitive radio. In: International Conference on Advanced Computer Control, pp. 212–216. IEEE, Singapore (2009)

7. Gosavi, A.: reinforcement learning: a tutorial survey and recent advances. Inf. J. Comput. **21** (2), 178–192 (2009)
8. Watkins, C.J.C.H., Dayan, P.: Machine Learning. Kluwer Academic Publishers, Dordrecht (1992)
9. Kang, S, Joo, C.: Combinatorial multi-armed bandits in cognitive radio networks: a brief overview. In: International Conference on Information and Communication Technology Convergence. IEEE, Jeju (2017)
10. Niimi, M., Ito, T.: Budget-limited multi-armed bandit problem with dynamic rewards and proposed algorithms. In: 4th International Congress on Advanced Applied Informatics, pp. 540–545. IEEE, Okayama (2015)
11. Das, A., Ghosh, S.C., Das, N.: Q-learning based cooperative spectrum mobility in cognitive radio networks. In: IEEE 42nd Conference on Local Computer Networks, pp. 502–505. IEEE, Singapore (2017)
12. Lo, B.F., Akyildiz, I.F.: Reinforcement learning-based cooperative sensing in cognitive radio ad hoc networks. In: 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 2244–2249. IEEE, Instanbul (2010)
13. Li, B., Yang, P., Wang, J.: Almost optimal dynamically-ordered channel sensing and accessing for cognitive networks. IEEE Trans. Mob. Comput. **13**(10), 1 (2014)