Dynamic Spectrum Access Based on Improved SARSA Algorithm

Xiaofei Liao, Dequan Wu* and Yang Wang
School of Information Science & Technology, Donghua University, Shanghai, China

*Corresponding author e-mail: wdq950510@163.com

Abstract. Dynamic spectrum access is the core technology of cognitive radio, accurate spectrum access can effectively alleviate the shortage of spectrum resources and realize the "secondary utilization" of spectrum resources. For this reason, this paper proposes a dynamic spectrum access based on improved Sarsa algorithm. The algorithm uses a simulated annealing strategy instead of the \( \epsilon \)-greedy strategy in action selection, which can effectively overcome the problem of the traditional Sarsa algorithm falling into a local optimum. Simulation experiments show that the algorithm proposed in this paper can effectively improve the throughput of cognitive users and reduce the probability of collision with the main user, what’s more ,it also has a significant increase in speed.

1. Introduction
In recent decades, with the rapid development of wireless communication technology and the extensive use of 3G and 4G mobile communication, especially 5G is coming soon, mobile communication has been popularized in all aspects of people's life. With the continuous development of science and technology, especially in the communication field, people's demand for wireless communication services has become stronger. In this respect, there is a contradiction between the spectrum that can be authorized to use in the existing spectrum resources and the growing service requirements. How to resolve their contradictions has become a key issue that needs to be solved urgently today. According to the previous report of the US Federal Communications Commission [1], it is known that in some large geographic areas, nearly 70% of the spectrum allocated through fixed policies is idle in most of the time. Therefore, the proposed of cognitive radio can solve the shortage of spectrum resources effectively, cognitive radio was proposed by Dr. Joseph Mitola of the Royal Swedish Institute of Technology in 1999[2]. Cognitive radio can perceive the surrounding wireless environment in a more flexible way, and intelligently interact with the external environment to ensure that authorized users are not affected and communicate normally. The cognitive user adjusts the access parameters in real time to adapt to the currently perceived wireless spectrum environment, thereby achieving spectrum sharing between the cognitive user and the authorized user[3]. Dynamic spectrum access is one of the key technologies for the development of cognitive radio networks. Cognitive users are required to understand the changes in the wireless network environment in time, so as to dynamically adjust the frequency band of the access network. Dynamic spectrum access technology uses the spectrum sensing and channel detection capabilities of cognitive radios to dynamically obtain the idle frequency band information of spectrum resources, and select idle frequency bands for use according to the sensing results, thereby improving the efficiency of spectrum resource use [4].
At present, due to the increasing demand for wireless mobile networks, the researches on cognitive radio dynamic spectrum access have attracted the attention of many scholars and experts. Therefore, a lot of research results have been born. In literature [5], a spectrum sensing scheme based on adaptive optimal support vector machine (SVM) is proposed. This algorithm can also obtain good detection probability in the case of low SNR, but the detection speed is very slow and the parameters are difficult to determine. For this reason, in literature [6], support vector machine classification model is used for time domain training and testing. It maps the input low-dimensional vector to the high-dimensional feature space by introducing a kernel function method. Literature [7] uses repeated games build reputations and apply punishments to achieve a desirable self-enforcing outcome. However, none of the above algorithms involve using the characteristics of the primary user and the channel environment to analyze the use of the spectrum. To this end, literature [8-10] proposes to use the hidden Markov model to model the cognitive environment. Due to the rapid change of the primary user, the cognitive user needs to negotiate frequently, which increases the overhead. Therefore, the literature [11] proposed the multi-user Q-learning algorithm for spectrum access, using the intelligence of the Q-learning algorithm to avoid conflicts due to lack of coordination and improve the accuracy of spectrum access. Currently, most of the researches on cognitive radio dynamic spectrum access are based on understanding the external environment, but in the actual communication, environmental modeling becomes very difficult due to the noise, channel interference and other factors.

In view of the above problems, this paper proposes a dynamic spectrum access based on improved Sarsa algorithm. Sarsa algorithm is a model-less reinforcement learning algorithm, which does not need to model the external environment, and it is an effective method for solving dynamic spectrum access. Because the traditional Sarsa algorithm uses the $\varepsilon$-greedy strategy, the results often fall into the local optimal solution. This paper introduces a simulated annealing strategy instead of the $\varepsilon$-greedy strategy, so that the spectrum access can obtain the global optimal solution and improve the accuracy of the spectrum access.

2. System Models

At present, with the occurrence of the AlphaGo event, artificial intelligence has become an upsurge of scientific researches. The core methods used in artificial intelligence are deep learning and reinforcement learning. Deep Learning is based on the study of artificial neural networks, it analyzes learning models by building neural networks which is similar to the human brain. Moreover, it maps simple underlying features to abstract high-level attributes, so it is possible to analyze and discover the characteristic properties of the data. Reinforcement learning algorithms provide a "trial and error" learning technique that tries various actions in the process of active interaction with the environment, and reflects the quality of the action through the size of the reward value obtained, thereby continuously updating the action strategy to reach the goal of learning optimal strategies[11].

2.1. Reinforcement Learning

Reinforcement learning algorithms refer to the individual who interact with the environment as agent. The agent constructs the state $s$ based on the perceived environment, and then selects action $a$ according to the strategy and executes it. When the selected action performs well, it will get a positive reward value from environmental feedback $r$, the agent will increase the probability of selecting the action $a$ again; when the action of the agent performing the selected action is not good, it will get a negative reward value of environmental feedback, and the agent will reduce the probability of choosing the action $a$ again. The following figure shows the interaction process between agent and environment in reinforcement learning:
2.1.1. Markov Decision Processes. For the dynamic spectrum access system, the primary users are authorized users who can access the spectrum at any time, the cognitive users can only periodically perceive whether the spectrum is occupied by the primary users. When cognitive users discover idle spectrum they will access it immediately, if they receive a signal that the primary users need to use this spectrum, they must quit at once. Assuming that the spectrum consists of N channels and is authorized to be used by the primary user, the channel bandwidth of each channel is \( B_i \), channel state \( S(t) = \{S_1(t), S_2(t), \ldots, S_N(t)\} \), channel state has two states: 0 (occupied) and 1 (idle), therefore, the state of the primary users using the spectrum can be modeled as a \( 2^N \) Markov process.

The Markov Decision Processes (MDP) consists of four elements <S, A, P, R>, where S represents the state set, A represents the action set, P represents the state transition probability, and R is the reward function. The dynamic execution process of the MDP is as follows: the initial state of the agent is s1, an action a1 is arbitrarily selected in the A action set, and the environment reward is r1. After execution, the agent moves to the next state s2 according to the state transition probability p, and then executes the next action a2... According to the above, we can get the following MDP execution process:

For a Markov decision process, the sum of rewards for a period is as follows:

\[
G_i = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{t-1} R_T
\]  

(1)

The goal of reinforcement learning is to find the optimal strategy to maximize the total reward. Due to the randomness of the environment, we can not determine whether the same reward can be obtained with the same action next time, therefore we add an attenuation factor \( \gamma \) to the reward, \( \gamma \in (0, 1) \), under a stable strategy \( \pi \), the value function of state s is defined as follows:

\[
V_\pi(s) = \mathbb{E}_\pi[G(t) | S_t = s] = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \ldots | S_t = s]
\]  

(2)

Considering that you can perform several actions in any state, introduce an action-value function \( Q(s,a) \) as follow:

\[
Q(s, a) = \mathbb{E}_\pi[G(t) | S_t = s, A_t = a]
\]  

(3)

For solving the reinforcement learning problem by the time difference(TD) method, the reward of one cycle is approximately as follows:

\[
G(t) = R_{t+1} + \gamma V(S_{t+1})
\]  

(4)
There is no complete sequence in the time series difference method, and the value function iteration formula of time series difference can be obtained as follows:

\[ V(S_t) = V(S_{t-1}) + \alpha(G(t) - V(S_{t-1})) \]  

(5)

In the above formula, \( \alpha \in (0, 1) \), the action value function at this time is:

\[ Q(s, a) = Q(s, a) + \alpha(G_t - Q(s, a)) \]  

(6)

2.1.2. Sarsa Algorithm. Sarsa algorithm is an on-policy time difference algorithm in reinforcement learning, it was proposed by Rummery and Niranjan in 1994 as a model-based algorithm[12]. Initially, it was called an improved Q-learning algorithm, and later it was renamed SARS (STATE-ACTION-REWARD-STATE-ACTION) learning algorithm[13]. The Sarsa algorithm has five major elements in the iteration process: <s, a, r, s', a'>. s, a represents the state action pair, r represents the reward value of (s, a), s' represents the transition state, and a' represents the sampling action of the transition state. The Sarsa algorithm selects Q(s, a) obtained by a in strict iteration according to the \( \varepsilon \)-greedy policy for iterative update, and the policy selection is consistent with the update process of Q(s, a). The update rule is expressed as:

\[ Q(s, a) = Q(s, a) + \alpha[R + \gamma Q(s', a') - Q(s, a)] \]  

(7)

In the above formula, \( \alpha \) represents the learning rate, and \( \gamma \) represents the discount factor. Sarsa algorithm has the advantage of fast process change in strategy evaluation. The convergence of the algorithm depends on the nature of the \( \varepsilon \)-greedy strategy, \( \varepsilon \)-greedy strategy is defined as follows:

\[
\pi(s_t) = \begin{cases} 
\arg \max_a Q(s, a) & \text{if } \delta < \varepsilon \\
\text{random} & \text{otherwise}
\end{cases}
\]  

(8)

In the above formula, \( \pi(s_t) \) is the strategy choice, \( \delta \in (0, 1) \) and \( \varepsilon \) is the greedy value. According to the Sarsa algorithm strategy selection analysis, it is known that the action is selected by comparing \( \delta \) and \( \varepsilon \), when the value of \( \varepsilon \) is large, there is a high probability that the action with a large Q value will be selected, the action with a small Q value will be randomly selected. Therefore, this makes the algorithm easily fall into a local optimal solution. The algorithm will eventually reach the optimal action, but the convergence speed is slow.

2.1.3. Simulated Annealing Strategy. The idea of Simulated Annealing (SA) was first proposed by Metropolis in 1953[13]. The simulated annealing algorithm generally consists of a double loop. In the inner cycle, multiple disturbances at the same temperature produce different states, and new states are accepted in accordance with Metropolis guidelines. According to the Metropolis criterion, the probability that a particle will reach equilibrium at temperature T is \( \exp(-\Delta E/KT) \), where E is the internal energy at temperature T, \( \Delta E \) is the number of changes in internal energy, and k is the Boltzmann constant. Metropolis formula is as follows:

\[
p = \begin{cases} 
\frac{1}{\exp\left(-\frac{E(X_{\text{new}}) - E(X_{\text{old}})}{T}\right)} & \text{if } E(X_{\text{new}}) < E(X_{\text{old}}) \\
\exp\left(-\frac{E(X_{\text{new}}) - E(X_{\text{old}})}{T}\right) & \text{if } E(X_{\text{new}}) \geq E(X_{\text{old}})
\end{cases}
\]  

(9)

In the outer loop, its contents include: temperature drop, algorithm iteration and judgment of convergence conditions. Its loop stop is controlled by the convergence condition. The simulated annealing algorithm will accept a solution that is worse than the current solution with a certain probability, so it may jump out of this local optimal solution and reach the global optimal solution. Take Figure 3 as an example, after searching the local optimal solution B, the simulated annealing algorithm will continue to move to the right with a certain probability. At this time, it is possible to jump out of the local best advantage B and reach the global best advantage C.
3. Dynamic spectrum access based on SA-Sarsa

3.1 SA-Sarsa Algorithm

The traditional Sarsa algorithm uses the $\varepsilon$-greedy strategy in strategy selection. In the greedy strategy, as the exploration continues, the cognitive user gets the Q value more and more tends to the optimal Q value. At this time, large-scale exploration will cause excessive system overhead and is not conducive to improving system performance. The spectrum access in this article is based on the Sarsa algorithm and the simulated annealing strategy is used as the basis of action iteration. The cognitive user selects a channel for spectrum listening according to the channel selection and access strategy, and obtains free/busy information of the channel by observing the wireless spectrum environment to make a channel access decision. Cognitive user gets rewards and interact with the environment. Therefore, the dynamic spectrum access process can be modeled as a finite Markov decision process. The mapping relationship based on the SA-Sarsa algorithm and the dynamic spectrum access of cognitive radio networks is shown in Table 1 below:

| SA-Sarsa Algorithm | Dynamic Spectrum Access |
|--------------------|-------------------------|
| Agent              | Cognitive User          |
| External Environment State Collection S | Set of channels accessible to cognitive users |
| Agent action set A | Set of channels the cognitive user is trying to access |
| The rewards the agent receives after taking action | Cognitive throughput of users after accessing the channel |
| Simulated annealing strategy | Cognitive user selects channel access |

In the above table, the return value obtained by the Agent after taking action is equivalent to the throughput obtained after the cognitive user accesses the channel. In the current state, the definition of the throughput obtained by the cognitive user in the $j$ time slot on the frequency band $b_i$ is:

$$R(j) = \frac{T_d}{T_d + T_s} I_i \phi_{T_d,j(i)} B_i$$

(10)

Among them, $T_d$ represents the length of data communication time in a complete time slot, and $T_s$ is the perceived access time of the channel. $I_i \in (0,1)$ is the sensing result of channel $i$, if the sensing result is busy $I_i = 0$ otherwise $I_i = 1$. In the transmission time slot $T_d$, $\phi_{T_d,j(i)}$ is the idle time of the frequency band $b_i$ in the $j$ time slot. $B_i$ is the bandwidth of the frequency band $b_i$. Cognitive user action selection adopts simulated annealing strategy, and controls the cognitive user's exploration rate through a temperature function to ensure the accuracy of cognitive users' access to idle frequency bands.
3.2. SA-Sarsa Algorithm Flow

From the analysis of the above algorithm, it can be known that the spectrum access based on the SA-Sarsa algorithm guides the cognitive user to select a channel through a return value, so that it selects a channel in the direction of maximizing user throughput. Cognitive users get any given action from the environment and access any channel. When the spectrum is found to be occupied, the environment returns a negative return value. When the accessed spectrum is found to be idle, the environment returns a positive return value. When the cognitive user detects a channel conflict, the corresponding Q value becomes smaller and smaller during the iteration process. According to the simulated annealing strategy, it can be seen that in the subsequent iteration process, the busy channel transmission service is avoided as much as possible. The algorithm flow is as follows:

Step 1: Given initial temperature $T$, cooling rate $q$

Step 2: Initialize the Q table $Q(s, a) \leftarrow (0, 0)$

Step 3: Initialization state $s$

Step 4: Simulated annealing process $T = T \times q$

Step 5: $s$ selects action $a$ under simulated annealing strategy

Step 6: Perform $a$, state transition to $s'$, get reward $R$

Step 7: $s'$ Select action $a'$ under annealing strategy

Step 8: Update Q table $Q(s, a) \leftarrow Q(s, a)$

Step 9: $s \leftarrow s'$, $a \leftarrow a'$

3.3. Simulation Results And Analysis

The simulation uses a comparison method, which is compared with the traditional Sarsa algorithm and random access strategy. In the random access strategy, the cognitive user arbitrarily selects a channel to listen with the same probability in each time slot. The parameters of the SA-Sarsa algorithm are set as follows: the number of channels is $n = 8$; the bandwidth of each channel is $B = 200$ kHZ; total length of each time slot $T_s = 100 \times 10^{-3} s$; to meet the listening requirements, set the listening time $T_s = 5 \times 10^{-3} s$; transmission time $T_d = 95 \times 10^{-3} s$; learning rate $\alpha = 0.8$; discount factor $\gamma = 0.9$. In the simulated annealing strategy, the initial temperature $T = 300$, the cooling rate $q = 0.95$, and the normal temperature is $T_{\text{min}} = 10^{-5}$. In order to be able to evaluate the overall average performance of the system, 50 time slots are combined into one long time slot (5s), and the throughput achieved by each long time slot is used as the evaluation index. It can be known from the simulation results in figure 4 that the SA-Sarsa algorithm has a large initial search rate and a small decrease in the search rate due to the high initial temperature in the early stage of the experiment, which makes the strategy randomly choose actions. Therefore, the search space is large in the first 10 long time slots, and the time step reward value is lower than the Sarsa algorithm. With the cooling process, at 15 long time slots, the search rate drops rapidly, the search efficiency increases significantly, and the convergence speed accelerates. The traditional Sarsa algorithm uses the $\epsilon$-greedy strategy in the iteration of the action, which causes the algorithm to converge slowly. It starts to converge at 26 long time slots and has poor stability. The SA-Sarsa algorithm started to converge at 17 long time slots, it achieves maximum throughput and good convergence stability, which made up for the shortcomings of traditional reinforcement learning with slow convergence speed and weak stability, and achieved good experimental results. It is further verified in figure 5 that the SA-Sarsa algorithm effectively reduces the collision probability and improves the stability compared with the traditional Sarsa algorithm.
4. Conclusion

Cognitive radio technology can alleviate the current shortage of spectrum resources, and traditional machine learning algorithms are difficult to apply to complex channel environments. To this end, this paper proposes the SA-Sarsa algorithm, without knowing the external environment information of spectrum access and the main user channel usage rules, by setting various parameters in the SA-Sarsa algorithm, cognitive users can accumulate experience in the continuous interaction learning process with the environment, and finally select the optimal channel access for communication. The simulation results show that the dynamic spectrum access based on SA-Sarsa algorithm can well improve the throughput of cognitive users and reduce the probability of collisions. What's more, it has fast convergence speed and good stability.

References

[1] Baldo N, Zorzi M. Cognitive network access using fuzzy decision making[J]. IEEE Transactions on Wireless Communications, 2009, 8(7).
[2] Mitola J, Maguire G Q. Cognitive radio: Making software measurements project summary [R]. Shared Spectrum Company Report, Aug. 2005.
[3] Haykin S. Cognitive radio: brain-empowered wireless communications[J]. IEEE journal on selected areas in communications, 2005, 23(2): 201-220.
[4] Song M, Xin C, Zhao Y, et al. Dynamic spectrum access: from cognitive radio to network radio[J]. IEEE Wireless Communications, 2012, 19 (1): 23-29.
[5] Zhuoran Cai, Honglin Zhao, Min Jia. Spectrum Sensing in Cognitive Radio Based on Adaptive Optimal SVM[J]. Information Technology Journal, 2011, 10(7).
[6] A. Subekti, H. F. Pardede, R. Sustika and Suyoto, "Spectrum Sensing for Cognitive Radio using Deep Autoencoder Neural Network and SVM," 2018 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), Serpong, Indonesia, 2018, pp. 81-85.
[7] Hongyan Li, Yang Liu and Dexian Zhang, "Dynamic spectrum access for cognitive radio systems with repeated games," 2010 IEEE International Conference on Wireless Communications, Networking and Information Security, Beijing, 2010, pp. 59-62.
[8] J. Wang, "Cooperative Spectrum Sensing in Cognitive Radio Networks Using Hidden Markov Model," 2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), Krakow, 2015, pp. 518-521.
[9] A. Raschellò, J. Pérez-Romero, O. Sallent and A. Umbert, "On the use of POMDP for spectrum selection in cognitive Radio networks," 8th International Conference on Cognitive Radio Oriented Wireless Networks, Washington, DC, 2013, pp. 19-24.
[10] J. A. Msumba and H. Xu, "Throughput optimization MAC scheme for cognitive radio networks: A POMDP framework," 2013 Africon, Pointe-Aux-Piments, 2013, pp. 1-5.
[11] H. Li, "Multi-Agent Q-Learning for Competitive Spectrum Access in Cognitive Radio Systems," 2010 Fifth IEEE Workshop on Networking Technologies for Software Defined Radio Networks (SDR), Boston, MA, 2010, pp. 1-6.

[12] Rummery, G A, Niranjan M. Online Q-learning using connectionist system[D]. London: Cambridge University, 1994.

[13] Sutton, R S, Barto A. G. Reinforcement learning: an introduction[M]. Cambridge: MIT, 1998: 150—185.