A FAST CONVERGENT CHANNEL SELECTION STRATEGY IN CRSN

Chun-mei Chen,*,** He-song Jiang,* Bin Wu,* Hong Jiang,* and Juan Zhang*

Abstract

Accurate and fast convergence to the optimal channel is a challenge in a cognitive radio sensor network (CRSN) when multiple cognitive wireless channels coexist. Some traditional wireless channel selection methods can be used to study the optimal channel selection. However, their convergence speed cannot meet the requirements because of vast computation and time accumulation. In this paper, a rapid channel selection strategy based on machine learning called MAB-CQ (multi-armed bandit-channel quality) is proposed. This strategy maps the channel selection problem to the improved multi-armed bandit (MAB) model. In the model, the second users (SUs) and the channels in the CRSN correspond to the players and the arms of MAB, respectively. The optimal channel is determined based on the UCB (upper confidence bound) of MAB-CQ for each player. In addition, the UCB equation is creatively defined to balance the exploration and exploitation problem. At the same time, to reduce the computation complexity, coefficients about the factors are used to narrow down the exploratory scope of our strategy. As a result, an accuracy optimal channel and a fast convergence speed are achieved by iterative execution of MAB-CQ. Extensive experimental results demonstrate that the MAB-CQ can converge to nearly 100% within the 10^5 time slots. By comparison, MAB-CQ has obvious advantages in cumulative rewards, computational complexity and convergence speed.

Key Words

Channel selection, multi-armed bandit, upper confidence bound, fast convergence, cognitive radio sensor network

1. Introduction

With the rapid development of wireless communication services, wireless sensor networks (WSNs) have been widely used, such as smart home, smart city, military, anti-terrorism, disaster relief, environmental monitoring and other fields [1]. WSN is composed of a large number of micro-sensor nodes, and the unlicensed spectrum is used between the communication nodes, such as the ISM (industrial scientific medical) band. However, with the number of devices using the unlicensed spectrum increasing exponentially, the network becomes seriously congested and the reliability of communication cannot be guaranteed. These elements greatly limit the development of WSN [2]. At present, cognitive radio technology is applied to WSN to form CRSN, which can alleviate the severity of the above problem. But, the CRSN also brings some challenges. For example, to improve channel access speed and selection accuracy is an urgent problem for SU [3]–[5]. Therefore, the research of channel selection in CRSN is of significance and it has become one of the hot research fields.

Cognitive radio technology is emerged to allow SUs for opportunistic access the spectrum holes when primary users are not active [6]. Thus, under the condition of multi-channel coexistence, it is crucial for SUs to make optimal decisions about which channel to access at different times. Recently, some scholars have researched them by machine learning [7]–[10]. Among them, MAB is a classical theory for selection. It includes two factors of exploration and exploitation and they need to be balanced [11]–[13].

From the above analysis, we can see that although many scholars have made some achievements in this field, there is still a lot of work to be done to improve the performance of CRSN, including the improvement of convergence speed and computational efficiency. In this paper, we focus on the improvement of the MAB method to select the optimal transmission channel. Through the reduction of exploration space and balancing the exploration and exploitation (E–E) problem, fast and accurate selection results can be obtained. Our contributions are summarized as follows.

- A state transition model based on the Gilbert–Elliott (G–E) Markov channel is constructed. In the condition that there is lack of prior information, the efficient channel selection method based on the model is researched in CRSN.
- A novel machine learning strategy named MAB-CQ is proposed to optimize the channel selection for SUs. On the basis of the classical MAB theory, we innovatively propose a solving equation containing three factors for UCB, and the optimal channel is determined based on UCB.

* School of Information Engineering, Southwest University of Science and Technology, Mianyang Sichuan 621010, China; e-mail: {ccm, jianghesong, wubin, jianghong, zhangjuan}@swust.edu.cn

** Institute of Electronic Engineering, China Academy of Engineering Physics, Mianyang Sichuan 621900, China

Corresponding author: Chun-mei Chen

Recommended by Prof. Anmin Zhu

(DOI: 10.2316/J.2020.206-0461)

- Furthermore, the coefficients of the exploration and channel quality estimation are put forward. Thus, MAB-CQ can resolve the E–E problem very well and improve the efficiency.
- Moreover, the time complexity and convergence speed are analysed. By comparing with other channel selection algorithms, our strategy MAB-CQ can curb inefficiency exploration, so it can use less execution time to converge faster.

The rest of the paper is organized in the following. The system model and some basic definitions of this paper are presented in Section 2. The novel channel selection strategy based on MAB is proposed in Section 3. In Section 4, the simulation results are analysed and the performances are evaluated. Finally, the conclusion of this paper is presented in Section 5.

2. System Model

In this paper, a CRSN model with multiple PUs and multiple SUs is considered. A work scenario is assumed as follows. Multiple PUs only represent multiple licensed channels in CRSN, and data transmission only occurs between SUs. Among these SUs, there is a destination SU and some source SUs with multiple sensor nodes. The destination SU is responsible for collecting all the information of source SUs. Each source SU collects their sensor data and sends them to the destination SU by the opportunity access licensed channel. If the source SU cannot directly reach the destination SU in one hop, the relay SUs are needed. So, in the process of working, the network perhaps includes multiple SU transmission pairs and each pair includes a transmitter SU and a receiver SU. We assume that each pair can opportunity access at least one channel in the valid range. Based on the above, we define a universal sensor network and the diagram is shown as Fig. 1. In this figure, the sensor nodes may be a temperature sensor or a humidity sensor or others according to the actual engineering needs.

In Fig. 1, SU-D is the destination SU and other SUs are sources. A PU represents one licensed channel. The solid circle shows the transmission range with a radius r_c and the dashed circle shows the sensing range with a radius r_s of SU. That is, only when one SU receiver is located in the transmission range of another SU transmitter, the SU transmission pair can link each other for reliable communications. Moreover, the transmission pair must select a common channel at common time slots. Here, common channel is the licensed channel within intersection area of transmission pair. For example, SU-D and SU_j is a transmission pair, and they can select the common channels PU_2, PU_3, PU_i or PU_n in the intersection area instead of the outside PU_1 and PU_{i+1}. Of course, SU will also face to select an available and optimal common channel for transmission when the multiple common channels coexist. On the contrary, if not any available common channels for the transmission pair, the relay SU will be adopted. For example, SU_3 will send data to SU-D, the SU_j will be as a relay SU if PU_3, PU_i and others are not available. In addition, we define $r_c \leq r_s$ because the available channels

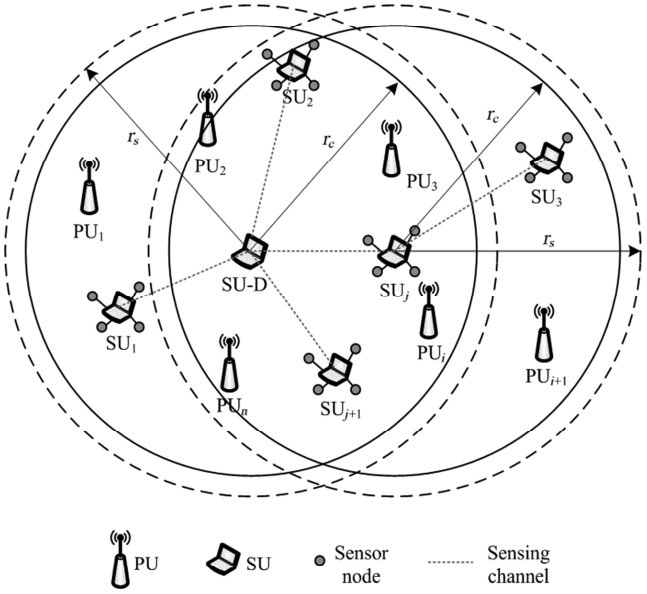

Figure 1. CRSN diagram.

Table 1
Main Parameters of System Model

Type	Description
$s=0$	Busy state of the current channel
$s=1$	Free state of the current channel
λ_0	The lower bound of the channel belief value
λ_1	The upper bound of the channel belief value

of an SU are based on the sensing range; each SU is not allowed to communicate with other SUs outside its sensing range because it may mistakenly use an occupied channel by a PU [14]. In this paper, we assume that the length of a slot is long enough to transmit a data packet.

We assume that there are N channels and each channel is mutually independent from others. The channel is modelled by the G–E Markov chain with two states: busy (denoted by 0) and free (denoted by 1), that is, the finite state space can be defined $\mathrm{S}=\{0,1\}$. If the channel is free, it allows SU to occupy. However, if the channel is busy, it will not be allowed to be used to avoid interfering with the current user. The channel state transition probabilities matrix can be expressed as follows:

$$P = \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix} = \begin{bmatrix} 1-\lambda_0 & \lambda_0 \\ 1-\lambda_1 & \lambda_1 \end{bmatrix} \tag{1}$$

and the state distribution can be expressed as follows:

$$p = [p_0, p_1] = \left[\frac{1-\lambda_1}{1-\lambda_1+\lambda_0}, \frac{\lambda_0}{1-\lambda_1+\lambda_0} \right] \tag{2}$$

These parameters are shown in Table 1 and Fig. 2. In (1), P_{ij} represents the channel state transition probability from i to j in two adjacent time slots, where $i, j \in \{0,1\}$

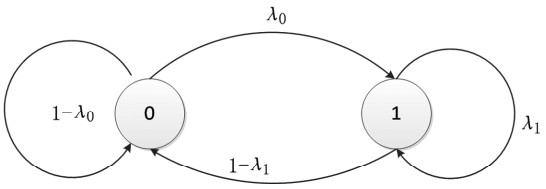

Figure 2. Channel state transition model.

Algorithm 1: Pseudo-code for Generate-states. This algorithm will predict each state for each channel based on channel transition probability and Markov Chain. λ_0 and λ_1 are global random matrixes. $TMAX$ and s are global variables. These parameters denote the network environment of the experiment in this paper.

Initialization:
Set the number of time slots $TMAX$;
Initial state array: $s_i[TMAX] = 0$;
Set random transition threshold $\vartheta[TMAX]$;
Output: s_i
Execute:
$\lambda_0 = lamda0[i]$;
$\lambda_1 = lamda1[i]$;
For $k = 1 : TMAX$
  istransfer=1;
  If ($s_i[k] == 0$ && $\vartheta[k] > \lambda_0$)
    istransfer=0;
  End If
  If ($s_i[k] == 1$ && $\vartheta[k] < \lambda_1$)
    istransfer=0;
  End If
  $s_i[k+1] = (1 - s_i[k]) * istransfer + s_i[k] * (1 - istransfer)$;
End For
End

Such as P_{01} represents the channel state transition probability from busy to free in two adjacent time slots. In (2), p_0 and p_1 represents the state distribution of busy and free, respectively. In this paper, we assume that λ_0 and λ_1 are the boundaries of belief values of channel, and the channel is positive correlated, which means $\lambda_0 \leq \lambda_1$.

Because the channel state cannot be observed directly, the next state of the system is completely calculated based on the current state according to the transition probability [15]. We construct the system model as the Markov chain. According to the Markov property, the next state is only related to the current state and not to other historical states. So, we can obtain all the channel states for each channel based on the state transition probability. We set a flag for the state transition condition, that is, mark 1 represents state transition, and mark 0 represents the non-transition. The pseudo-code is shown in Algorithm 1. In the algorithm, $\vartheta \subset (0, 1)$ and the variable "istransfer" is flag.

Although there may be multiple licensed channels available in CRSN, the channel quality is probably variable with the change of environment. It brings trouble for transmission channel selection. Surely, to maximize the transmission performance, we expect to choose the optimal channel in each time slot. In theory, an effective method can gradually approximate this ideal state. However, to avoid excessive overhead caused by repeated channel handoff, we usually choose the optimal channel in the long run based on a large amount of statistical information. The example diagram is shown as Fig. 3.

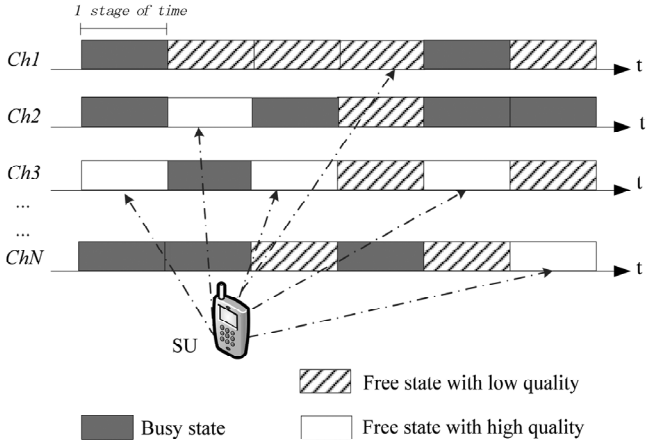

Figure 3. Channel quality information and opportunistic access diagram.

In Fig. 3, there are N licensed channels and SU will opportunity access one channel among them at each time slot or in 1 stage of time. In each time axis, there are many rectangles and a rectangle represents a time slot or a stage of time. Different fillers in rectangles show the different channel states. They are "free state" with white filled and "busy state" with dark filled. Among them, "free state with high quality" is filled with pure white, and "free state with low quality" is filled with the slash. With the support of the selection strategy, the system always selects the optimal channel for transmission in each time column. For example, at the first time column, SU selects the "free state with high quality" Ch3 but not the other busy channels. At the second time slot column, SU selects the "free state with high quality" Ch2, and it avoids the "free state with low quality" Ch1 and other busy channels. At the third and fifth time slot columns, SU always selects the Ch3. It is thus clear that, from the finite visible horizon time axis and channels, Ch3 has the most selection times. It is because Ch3 has the most time column in "free state" and keeps "high quality", and it has the most opportunities to be selected. So, it is considered as the optimal channel.

In this paper, we consider the influence of channel state on reward. We assume that when the channel state is busy, the transmission will be failure and a penalty $R_c < 0$ will be imposed. When the channel state is free, the transmission will be success and an award $R_r > 0$ will be given. According this, we define the instantaneous actual reward as follows:

$$\chi_i(t) = \begin{cases} R_r & \text{if the channel is free} \\ R_c & \text{if the channel is busy} \end{cases} \tag{3}$$

where $\chi_i(t)$ is the actual reward of channel i at time slot t.

3. Channel Selection based on multi-armed bandit (MAB)

We assume that SU can access at least one available channel from N channels, but the gains from different channel are unknown. So, we can define the process of SU selecting the channel as an MAB problem. Based on the classic MAB theory [11], we refer to each channel as an arm separately, and SU as a player. SU will obtain some gains after playing one arm (namely "access a channel"). Our goal is to find the optimal arm in a finite time and to maximize the total gains in all time slots.

3.1 Upper Confidence Bound

In the MAB model, the player will receive some rewards at each slot if one arm is chosen. Assuming after n time slots, channel i has been chosen $T_i(n)$ times by SU. Then the expected gains mean can be expressed as follows:

$$u_i(n) = \sum_{k=1}^{T_i(n)} \chi_i(k) \Big/ T_i(n) \tag{4}$$

where $\sum_{k=1}^{T_i(n)} \chi_i(k)$ denotes the total gains after channel i is selected $T_i(n)$ times in n slots. Here, each arm at least is played once and $T_i(n) \geq 1$. Then, $u_i(n)$ are the average gains for each channel i. In [17], Chen *et al.* proposed the classical strategy UCB1 and rigorously derived to balance for the exploitation and exploration. In view of the above, we can obtain the expected upper confidence bound:

$$UCB_i(n) = u_i(n) + \sqrt{2\ln(n)/T_i(n)} \tag{5}$$

where $u_i(n)$ is the exploitation factor and $\sqrt{2\ln(n)/T_i(n)}$ is the exploration factor. The exploration factor is used to explore other arms, prompting new choices not to be too rigid with the performance of selected arms. With the increasing times of channel i selected, the exploration factor will decrease, but the average gains will increase. When the channel i is selected for enough times, the ratio of numerator and denominator tends to the smallest, and the expected average gains are mainly determined by $u_i(n)$. It can be seen that less exploration may lead to local optimum; more exploration may increase the cost and hinder the performance of the algorithm. Therefore, this paper attempts to adjust the weight of exploration factor to solve this problem better.

UCB1-tuned is an enhancement suggested by Auer *et al.* [11] to tune the bounds more finely. Based on their ideas, we replace the exploration factor $\sqrt{2\ln(n)/T_i(n)}$ of (5) with

$$\xi_i(n) = \sqrt{\frac{\ln(n)}{T_i(n)} * \min\left[\frac{1}{4}, \varphi_i(n)\right]} \tag{6}$$

where $\varphi_i(n)$ represents a deviation factor associated with the variance of channel i. It can reflect the fluctuations about a series of instantaneous gains of channel i. It will dynamically adjust the exploration interval of the sub-optimal solutions and reduce the cost of exploration. The expression is

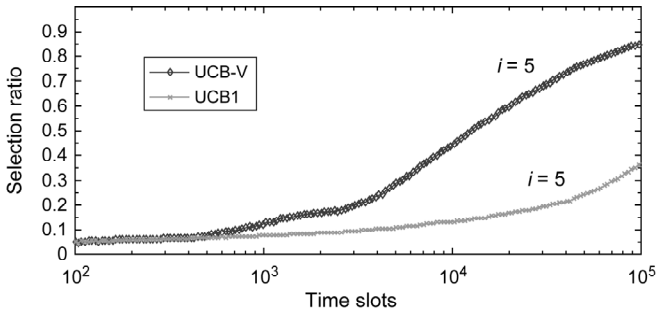

Figure 4. Convergence speed comparative (x-axis is the logarithmic coordinate).

$$\varphi_i(n) = \delta_i^2(n) + \sqrt{2\ln(n)/T_i(n)} \tag{7}$$

where $\delta_i^2(n)$ is about the instantaneous gains variance of channel i. The instantaneous gains mean (the sum of square of instantaneous gain of channel i at each time be divided by total times selected) subtracts the square of actual empirical gains mean of channel i. The expression is

$$\delta_i^2(n) = \frac{\sum_{k=1}^{T_i(n)} \chi_i^2(k)}{T_i(n)} - u_i^2(n) \tag{8}$$

where $\sum_{k=1}^{T_i(n)} \chi_i^2(k)/T_i(n)$ means arithmetic mean about the instantaneous gains. $u_i^2(n)$ represents the square of the average gains obtained by selecting channel i $T_i(n)$ times in n time slots. Based on our previous research [16], [17], we can obtain the upper confidence bound $g_i(n)$ as

$$\begin{aligned} g_i(n) &= u_i(n) + \xi_i(n) = u_i(n) + \sqrt{\frac{\ln(n)}{T_i(n)} * \min\left[\frac{1}{4}, \varphi_i(n)\right]} \\ &= u_i(n) + \sqrt{\frac{\ln(n)}{T_i(n)} * \min\left\{\frac{1}{4}, \left[\frac{\sum_{k=1}^{T_i(n)} \chi_i^2(k)}{T_i(n)} - u_i^2(n) + \sqrt{\frac{2\ln(n)}{T_i(n)}}\right]\right\}} \end{aligned} \tag{9}$$

We called this method as UCB-V (UCB-Variance) in this paper.

Through UCB-V, the next channel to be chosen will be determined by the value of the current $g_i(n)$. Based on the Bellman equation, we can obtain the optimal channel i^* as

$$i^* = \underset{i \in N}{\operatorname{argmax}}(g_i(n)) \tag{10}$$

To test the effect of the exploration factor, the convergence speed between UCB1 and UCB-V are compared. Under the same conditions (scenario 1 in chapter 5), the two methods choose their own optimal channel from 20 channels, respectively. From Fig. 4, we can see that they are all selecting the optimal fifth channel, but the selection ratio of UCB-V goes up faster than that of UCB1. It means that UCB-V converges faster than UCB1. This is because the new exploration factor $\xi_i(n)$ optimizes the scope of exploration and increases the speed of exploration.

3.2 Optimization Strategy

From the UCB-V strategy mentioned above, we can see the exploration time can be optimized. In this section, to further optimize the scope of exploration, a new exploration factor associated with channel quality is considered. It is easy to know, if the channel state is always free, *i.e.*, $s_i(k) = 1, \forall k \in T_i(n)$, then we can approximate that the quality of channel $i$is very good. So, based on channel states, a confidence factor about channel quality of channel i is defined as

$$G_i(n) = \frac{\sum_{k=1}^{T_i(n)} s_i(k)}{T_i(n)} \tag{11}$$

where $G_i(n)$ is the confidence factor of channel quality and it represents the exploitation contribution for channel i. In addition, the ideal maximum value of expected confidence factor within the set of channel states is defined as $G_{\max} = \max_{i \in N}(G_i(n)) = 1$. Then, the quality gap of each channel i is defined as

$$\Delta G_i(n) = G_{\max} - G_i(n) \tag{12}$$

According to the analysis in Fig. 3, the bigger the G_i is, the smaller the ΔG_i is and the better the channel quality is. So, we should select this channel with little gap for transmission.

To balance the quality confidence and exploration degree, two coefficients α and β are defined as the exploration coefficient for arms and channel quality, respectively. So, we obtain an improved equation and achieve the new upper confidence bound g'

$$g_i'(n) = u_i(n) + \alpha * \xi_i(n) + \beta * G_i(n) \tag{13}$$

In (13), α and β are the weight coefficients about exploration factor and confidence of channel quality, respectively. If either or both of α and β are increased, $g_i^{'}$ will increase, which means that we trust the current channel i more and it has a greater chance of being selected. On the contrary, if α and β decrease, we will explore other more channels for better quality and availability. So, formula (10) of selecting the optimal channel can be rewritten as

$$i^{*'} = \underset{i \in N}{\operatorname{argmax}}(g_i'(n)) \tag{14}$$

In this paper, the objective function is (14) and g_i' is the upper confidence bound of the channel i and it can measure the statistical gains of channels. That is, the channel with greatest statistical gains is called the optimal channel and it will be selected next time. But, the strategy's overall performance is also measured by parameter "regret" in the MAB machine learning field. The strategy should aim to minimize the regret. Here, we define the regret is: after n time slots, the deviation between the best ideal expected reward and the actual reward. The expression of the total regret value is

$$R(n) = \sum_{t=1}^{n}(\chi_{opt} - \chi_t) = n\chi_{opt} - \sum_{t=1}^{n}\chi_t \tag{15}$$

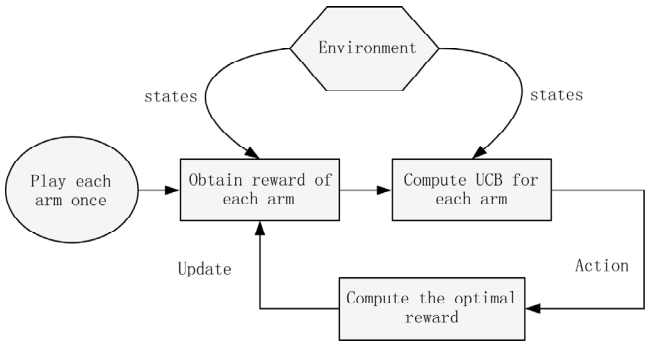

Figure 5. Learning model of MAB-CQ.

where χ_{opt} is the best expected reward in all channels. χ_t is the immediate reward of the selected channel at slot t and it may be different because of different channel selected. So, the second item $\sum_{t=1}^{n}\chi_t$ denotes the actual accumulation gains in n time slots. $\mathcal{R}$ is the expected loss due to the policy maybe not always get the best expected reward at each time slot. The higher the regret value is, the more unsuccessful the channel selection optimization algorithm is. So, the regret value should be as low as possible.

3.3 Algorithm Description

Based on the analysis of above, the optimal value of UCB determines the next selected channel. Through our optimization, g has been improved from two parts to three parts of g'. Accordingly, a novel channel selection optimization algorithm MAB-CQ is proposed. The machine learning principle of the algorithm is as follows. First, it generates the G–E channel states for each time slot of each channel based on the Markov chain and then starts to access all channels once and obtain some initial values, such as reward and selected times. Next, it loops execution for "computation-selection-update-computation". After each loop, MAB-CQ policy updates the reward once for the current arm with maximum upper confidence bound g'. After many cycles learning, this strategy will converge to the optimal channel finally. The learning model is shown in Fig. 5, and the pseudo-codes are described in Algorithms 2 and 3.

In Fig. 5, we will obtain the initial reward value for each arm by "Play each arm once". The "Environment" represents channel states and state transition probabilities and it will act on the calculation of channel reward and upper confidence bound. The "Action" will select one best arm in current time and calculate its reward to update the initial reward. Then, the algorithm enters an iteration period to search for the better one. So, we can see that the strategy is not limited to the current best arm, but explores for more arms through loops learning to find global optimal arm.

To analyse the time performance of our strategy, the time complexity of six learning algorithms is compared. The six algorithms aim to study the optimal channel selection. They are Q-learning [18], RCA [19], RQoS-UCB [13], UCB [9], ε – Greedy [9] and our strategy MAB-CQ.

Algorithm 2: Pseudo-code for Get-rewards. This algorithm will count the rewards R_i of selected channel i, and T_i is the number of times selected; n is the total number of times for all channels selected.

Initialization:
$R_r = 2, R_c = -0.5$;
Output: R_i, T_i, n
Execute:
If current state is 0
  $\chi_i = R_c$;
Else
  $\chi_i = R_r$;
End if
$R_i = R_i + \chi_i$;
$T_i = T_i + 1$;
$n = n + 1$;
End

Algorithm 3: Pseudo-code for MAB-CQ. This is the core algorithm in our paper.

Initialization:
Set arm numbers N;
Set $\alpha > 0, \beta > 0$;
For i=1:N /* Play each arm once and obtain the initial values */
  Generate-states (i);
  Get-rewards(i);
End For
Output: The optimal arm $i^{*'}$
Loop: /* Loop execution until convergence */
While ts < TMAX
  For i=1:N /* Calculate the upper confidence bound for each arm*/
    $u_i(n) = R_i(n)/T_i(n)$;
    $\xi_i(n) = \sqrt{\frac{\ln(n)}{T_i(n)} * \min\left\{\frac{1}{4}, \left[\frac{\sum_{k=1}^{T_i(n)} \chi_i^2(k)}{T_i(n)} - u_i^2(n) + \sqrt{\frac{2\ln(n)}{T_i(n)}}\right]\right\}}$;
    $G_i(n) = \frac{\sum_{k=1}^{T_i(n)} s_i(k)}{T_i(n)}$;
    $g_i^{'}(n) = u_i(n) + \alpha * \xi_i(n) + \beta * G_i(n)$;
  End For
  $i^{*'} = \underset{i \in N}{\operatorname{argmax}}(g_i^{'}(n))$;
  Get-rewards($i^{*'}$);
  $ts = ts+1$;
End While
End

The results are summarized in Table 2. Our comparison is based on the pseudo-code description of the six algorithms in the corresponding literature. The execution frequency is a function $f(*)$ on the scale of the problem, where * denotes the symbol of problem scale. That is, it is the sum of the operation times or the total number of times executed about each core statements in the algorithm Time complexity is an O function related to the function $f(*)$, it means to take an expression with the highest power about the problem scale in $f(*)$. In addition, it can be expressed as $T(*) = O(f(*))$. From Algorithm 3, we can obtain the execution frequency is about $f(KN) = K(2N + 3) + N$. The calculation method is as follows: the loop count in lines 3–6 is about $N(K + 1)$, and the loop count in lines 7–17 is about $K(N + 3)$. So the cumulative number is about $K(2N + 3) + N$, and the time complexity is $T(KN) = O(f(KN)) = O(NK)$. Here, K and N are the number of time slots and channels, respectively. The parameter D is defined as the number of iterations in [9] and usually it is a big constant of the same order of magnitude as the time slots K. From "Time complexity" in Table 2, the values of Q-learning, RCA, RQoS-UCB and our strategy MAB-CQ are the same, that is $O(KN)$. But from "Execution frequency" in Table 2, our strategy MAB-CQ performs fewer operations. When $D \gg N$, UCB and ε − Greedy policies execute slowly, and our strategy MAB-CQ has the best "Execution frequency" among the six strategies. As seen in numerical analysis, it is clear that the execute speed of our strategy MAB-CQ is faster.

Table 2
Algorithm Complexity

Leaning Algorithm	Execution Frequency	Time Complexity
Q-learning	K($6N$+3)	O(KN)
RCA	K($3N$+3)	O(KN)
RQoS-UCB	K($8N$+6)	O(KN)
UCB	KD(N+7)	O(KND)
ε − Greedy	KD(N+13)	O(KND)
MAB-CQ	K($2N$+3)+N	O(KN)

Table 3
The Parameters in the Simulation

Parameters	Value
Number of channels	$N = 20$
Time slots	$n = 10^5$
Award	$R_r = 2$
Penalty	$R_c = -0.5$
Exploration coefficient	$\alpha = 0.7$
Confidence coefficient	$\beta = 0.3$

4. Simulation and Performance Analysis

In this section, we test our strategy under many different scenarios. To facilitate the analysis, we chose two representative scenarios from these experiments to elaborate the experimental results. The main parameters of experiments are shown as Table 3. In addition, coefficient α and β can

Table 4
Channel Parameters for Two Scenarios

Channel	Scenario 1					Scenario 2				
No.	λ_0	λ_1	p_{01}	p_{10}	Select Times	λ_0	λ_1	P_{01}	P_{10}	Select Times
1	0.09	0.68	0.09	0.32	92	0.25	0.92	0.25	0.08	2,043
2	0.67	0.82	0.67	0.18	1,513	0.23	0.89	0.23	0.89	1,139
3	0.15	0.55	0.15	0.45	105	0.58	0.70	0.58	0.30	344
4	0.59	0.66	0.59	0.34	374	0.29	0.50	0.29	0.50	58
5	0.83	0.92	0.83	0.08	3,988,728	0.31	0.71	0.31	0.29	255
6	0.46	0.60	0.46	0.40	340	0.18	0.47	0.18	0.53	48
7	0.60	0.71	0.60	0.29	361	0.59	0.78	0.59	0.22	638
8	0.37	0.63	0.37	0.37	228	0.52	0.78	0.52	0.22	499
9	0.24	0.85	0.24	0.15	1,273	0.21	0.65	0.21	0.35	158
10	0.15	0.38	0.15	0.62	46	0.57	0.66	0.57	0.34	271
11	0.27	0.32	0.27	0.68	80	0.80	0.98	0.80	0.02	3,988,933
12	0.26	0.61	0.26	0.39	128	0.24	0.94	0.24	0.06	3,760
13	0.22	0.70	0.22	0.30	188	0.14	0.19	0.14	0.81	54
14	0.16	0.82	0.16	0.18	573	0.31	0.49	0.31	0.51	80
15	0.20	0.51	0.20	0.49	117	0.63	0.82	0.63	0.18	882
16	0.19	0.84	0.19	0.16	767	0.53	0.63	0.53	0.37	268
17	0.04	0.94	0.04	0.06	2,844	0.32	0.47	0.32	0.53	80
18	0.29	0.60	0.29	0.40	129	0.15	0.35	0.15	0.65	71
19	0.50	0.72	0.50	0.28	585	0.53	0.67	0.53	0.33	292
20	0.66	0.83	0.66	0.17	1,529	0.15	0.75	0.15	0.25	127

be adjust, the $\lambda 0$ and $\lambda 1$ are randomly generated arrays whose values are shown in Table 4.

In Table 4, the elements about channel state transition probabilities matrix P_{01} and P_{10} are computed based on Section 3. The select times of each channel are computed based on Section 4.

In Table 4, the selection times are counted about 20 channels in two different scenarios. About the run time, we first set the time slot value 10^6 as a round and then loop four times. In scenario 1, we can see that channel 5 has the most select times with 3,988,728. In scenario 2, we can see that channel 11 has the most select times with 3,988,933. They are far more than the select times of other 19 channels. Thus, our algorithm MAB-CQ chooses the optimal channel is No. 5 under scenario 1, while it is No. 11 under scenario 2.

Next, we verify the convergence of MAB-CQ in these two different scenarios. Fig. 6 shows the variation curve of the percentage of channels selected with increasing time slots in two scenarios. At a certain time slot, the higher the curve horizontal location is, the more times the corresponding arm is chosen; otherwise, the arm is the less

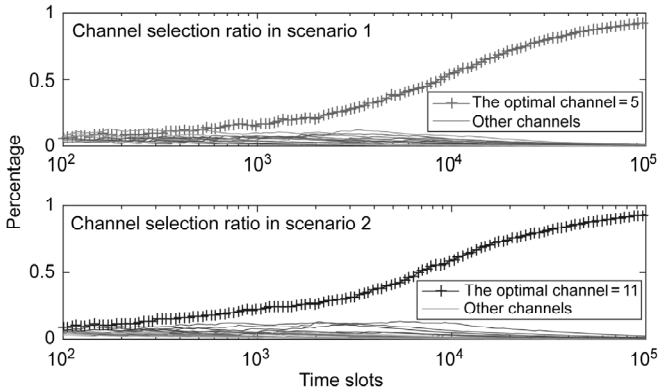

Figure 6. Channel selection ratio in two scenarios (x-axis is the logarithmic coordinate).

chosen. We can see from Fig. 6, about before 10^3 time slots, there is no much difference in the percentage of 20 arms. This is because all arms probably are selected in machine learning period, and at this time, their confidences are all very low. Almost from the beginning of 10^3 time

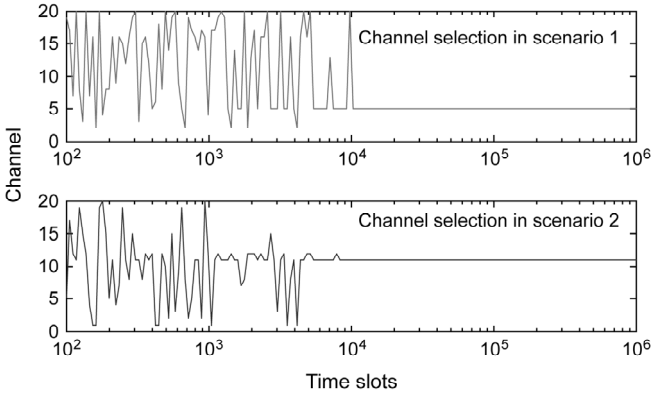

Figure 7. Channel selection in two scenarios (x-axis is the logarithmic coordinate).

slot, the 5-th curve in the upper branch and the 11-th curve in the lower branch of Fig. 6 are rising and they will gradually reach to 100% in the long run. That means that the optimal channels in the two scenarios are found separately. From Table 4, we can also see that the 5-th channel is the optimal channel in scenario 1, and the 11-th channel is the optimal channel in scenario 2. It shows that the corresponding optimal arm is the same in Table 4 and Fig. 6.

To further verify the correctness of MAB-CQ, we carry out the optimal channel selection experiments in scenario 1 and scenario 2 from another aspect. They are shown as the upper branch and the lower branch of Fig. 7, respectively. It is easy to see, after about 10^4 time slots, MAB-CQ finally converges to the 5th channel and the 11th channel, respectively. As we hope, this conclusion is consistent with Table 4 and Fig. 6. From Fig. 7, we can see that MAB-CQ can converge to nearly 100% after 10^5 time slots, no matter scenario 1 or scenario 2.

According to above experiments, it is enough to run 10^5 time slots for convergence of our algorithm. To facilitate analysis and display, we sample every 500 time slots at intervals and divide 10^5 time slots into 200 samples. To verify the stability of MAB-CQ, we will calculate some statistics in every 500 time slots, including statistical averages of rewards and regrets respectively. The experimental results are shown in Figs. 8 and 9.

Figure 8 shows the fluctuations about actual average reward in scenario 1 and scenario 2, respectively. Here, we define the maximum instantaneous reward in 20 channels as the ideal reward. That is, the maximum instantaneous reward is $\chi_{opt} = 2$ in our experiments. In this part, we test the performance of MAB-CQ by the fluctuation of the actual reward nearby the ideal reward. It can be seen from Fig. 8 that the average reward is not same each other at every sample point. This is because the states of these channels are variable and their transition probabilities are different. So, in different sampling intervals, their rewards are different which leads to fluctuations. But, these fluctuations are in a smaller range and they are gradually narrowing with the time. Compare scenario 1 with scenario 2, the trend of average reward in scenario 2 is quickly closer to the ideal reward, which shows that the

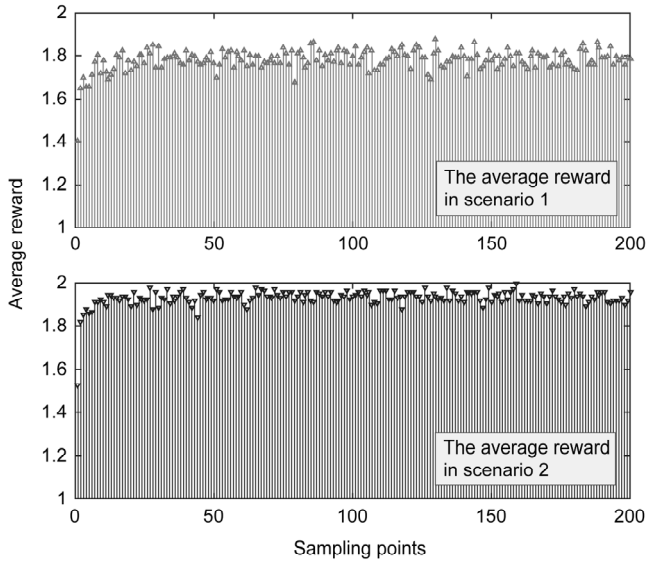

Figure 8. The fluctuation of the actual average reward (x-axis is the linear coordinate).

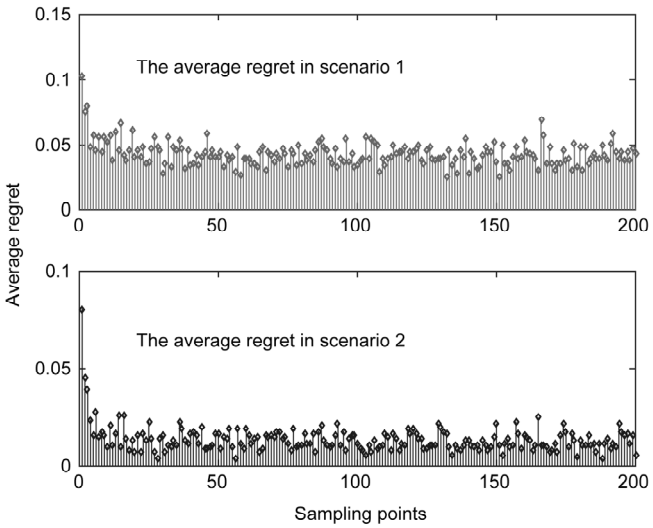

Figure 9. The fluctuation of the actual average regret (x-axis is the linear coordinate).

statistics rewards of scenario 2 is also better than that of scenario 1.

Figure 9 is the fluctuation of the actual average regrets in scenario 1 and scenario 2, respectively. Here, the ideal regret value should be zero under the condition of ideal reward. But, it can be seen from Fig. 9 that the curves have some fluctuations. The reason is similar to Fig. 8. If the reward increases, the regret decreases, and vice versa. From these fluctuation curves, the trend of average regrets in scenario 2 is closer to 0, which shows that the statistics regrets of scenario 2 are better than that of scenario 1. This conclusion is echoed with Fig. 8.

Even better, we change the channel parameters for many times and repeat the experiments and the selected result is still following the corresponding optimal arm. From Figs. 6 to 9, MAB-CQ can find the optimal arm in different scenarios. In conclusion, MAB-CQ has good accuracy, reliable and robustness.

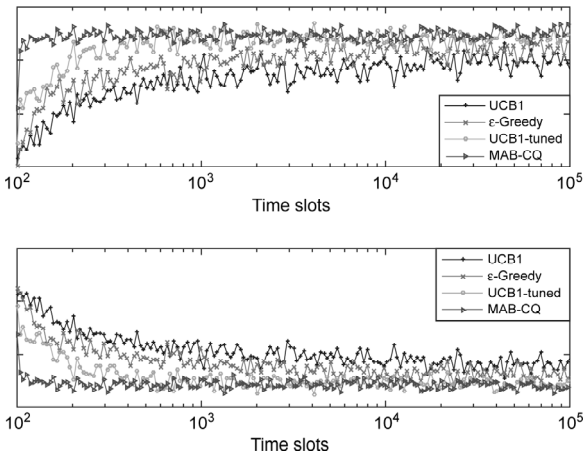

Figure 10. Cumulative rewards and regrets comparisons (x-axis is the logarithmic coordinate).

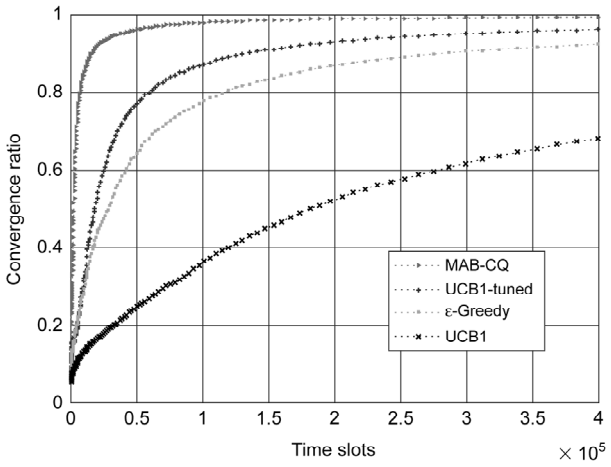

Figure 11. Convergence speed comparison (x-axis is the linear coordinate).

Finally, to verify the advantage of MAB-CQ, we do some comparative experiments with UCB1, $\varepsilon-$Greedy and UCB1-Tuned. The results are shown in Figs. 10 and 11. The cumulative rewards and cumulative regrets in scenario 1 are shown in Fig. 10. We still divide each 500 time slots into one statistical stage. From the upper branch of Fig. 10, the earliest converges to the optimal cumulative rewards is our policy MAB-CQ, *i.e.*, from the y-axis, the curve of MAB-CQ exceeds 800 at the earliest time among four policies, UCB1-Tuned is next and it approached MAB-CQ earlier than other two strategies. That is because the instantaneous gains variance is used for optimal channel selection in both MAB-CQ and UCB1-Tuned strategies, which will make them faster to find good channels and thus achieve high cumulative rewards than other two strategies. Besides, the confidence of channel quality is also used in MAB-CQ, so it convergences faster and obtains high rewards earlier than UCB1-Tuned. From the lower branch of Fig. 10, the cumulative regrets of four strategies have the opposite order of values on the y-axis as compared with the upper branch of Fig. 10. This is because rewards and regrets are mutually constrained, and this conclusion is in line with the analysis about the upper branch of Fig. 10.

In this paragraph, the execution speed is compared among $\varepsilon-$ Greedy, UCB1, UCB1-Tuned and Our MAB-CQ. From Fig. 11, we can see that they all can convergence to 100% in the long run. However, the time required for the convergence ratio to reach 100% is quite different from each other. First, UCB1 may take a long time to determine. Second, $\varepsilon-$ Greedy and UCB1-Tuned are better than UCB1, but they both converge to 100% over $4*10^5$ slots. Finally, our MAB-CQ is the fastest convergent curve and it is close to 100% at $0.5*10^5$ time slot. From these data, we know the speed of MAB-CQ is the fastest than others and UCB1 is the slowest among them. The reason is that four strategies use different principles to explore channels, *i.e.*, UCB1, UCB1-Tuned, $\varepsilon-$ Greedy only explore channels based on the current reward, just only their exploration weights are different. However, our policy MAB-CQ takes into account the channel quality confidence in addition to channel availability. Hence, MAB-CQ allows offering higher executing efficiency than the other strategies and it has obvious advantages in convergence ratio.

5. Conclusion

An efficient policy named MAB-CQ is proposed about channel selection in CRSN without sufficient prior knowledge of channels. It is an innovative strategy based on the MAB model, and it adds the confidence factor of channel quality on the basis of the traditional MAB. The optimal channel solution based on the improved UCB is realized. Through machine learning and channel quality estimation, the strategy can quickly converge to the optimal channel and its time complexity is lower than other algorithms. MAB-CQ solves the problems about fast convergence and accurate channel selection in CRSN when multiple cognitive wireless channels coexist. The experimental results show that MAB-CQ is stable and reliable.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China with Grant No. 61771410.

References

[1] Z. Ren, G. Zhang, D. Lin, Z. Zhang, and X. Zhao, Review on application of WSNs, *Transducer and Microsystem Technologies*, *37*(3), 2018, 1–2.

[2] Y. Jin, Research on cooperative spectrum sensing technology in cognitive wireless sensor networks (Master's thesis), Nanjing: Nanjing University of Posts and Telecommunications, 2015 (in Chinese).

[3] M. Yang, Design and implementation of wireless sensor network system based on cognitive radio, *Fire Control and Command Control*, *41*(11), 2016, 182–186 (in Chinese).

[4] S. Bayhan and Z.F. Alag, A Markovian approach for best-fit channel selection in cognitive radio networks, *Ad Hoc Networks*, *12*, 2014, 165–177.

[5] B. Ma, X. Bao, and X. Xie, Spectrum handoff algorithm in cognitive radio networks: A survey, *ACTA Electronic Sinica*, *44*(6), 2016, 1496–1503.

[6] D. Darsena, G. Gelli, and F. Verde, An opportunistic spectrum access scheme for multicarrier cognitive sensor networks, *IEEE Sensors Journal*, *17*(8), 2017, 2596–2606.

[7] B. Han, H. Jiang, Y. Luo, and J. Zhou, Cognitive radio resource allocation based on the improved quantum genetic algorithm, *International Journal of Robotics and Automation*, *34*(4), 2019, 451–460.
[8] J. Ni, X. Li, M. Hua, and S.X. Yang, Bioinspired neural network-based Q-learning approach for robot path planning in unknown environments, *International Journal of Robotics and Automation*, *31*(6), 2016, 4526–4590.
[9] H. Chen, Research on channel selection mechanism based on multi-armed bandit in cognitive network (Master's thesis), Chongqing: Chongqing University of Posts and Telecommunications, 2016 (in Chinese).
[10] X. You, X. He, X. Han, C. Wu, and H. Jiang, Cross-layer parameters reconfiguration in industrial cognitive wireless networks using Moabchv algorithm, *International Journal of Robotics and Automation*, *33*(2), 2018, 150–160.
[11] P. Auer, N. Cesa-Bianchi, and P. Fischer, Finite-time analysis of the multiarmed bandit problem, *Machine Learning*, *47*(2–3), 2002, 235–256.
[12] S.Q. Yahyaa, M.M. Drugan, and B. Manderick, Exploration vs exploitation in the multi-objective multi-armed bandit problem, *IEEE 2014 Int. Joint Conf. on Neural Networks*, Beijing, China, 2014, 2290–2297.
[13] N. Modi, P. Mary, and C. Moy, QoS driven channel election algorithm for cognitive radio network: Multi-user multi-armed bandit approach, *IEEE Transactions on Cognitive Communications and Networking*, *3*(1), 2017, 49–66.
[14] Y. Song and J. Xie, Distributed broadcast protocol with collision avoidance in cognitive radio ad hoc networks, *Broadcast Design in Cognitive Radio Ad Hoc Networks*, (Springer, Cham, 2014), 37–65.
[15] Y. Wu and B. Krishnamachari, Online learning to optimize transmission over an unknown Gilbert–Elliott channel, *IEEE Int. Symp. on Modeling & Optimization in Mobile*, Paderborn, Germany, 2012, 27–32.
[16] Z. Juan, J. Hong, H. Zhenhua, C. Chunmei, and J. Hesong, Study of multi-armed bandits for energy conservation in cognitive radio sensor networks, *Sensors*, *15*(4), 2015, 9360–9387.
[17] C. Chen, B. Wu, X. Tuo, H. Jiang, and J. Zhang, Relay translation policies based on state pruning for ad hoc networks, *International Journal of Robotics and Automation*, *33*(3), 2018, 247–257.
[18] X. Chen, Z. Zhao, and H. Zhang, Stochastic power adaptation with multiagent reinforcement learning for cognitive wireless mesh networks, *IEEE Transactions on Mobile Computing*, *12*(11), 2013, 2155–2166.
[19] C. Tekin and M. Liu, Online learning in opportunistic spectrum access: A restless bandit approach, *2011 Proc. IEEE INFOCOM Conf.*, Shanghai, China, 2011, 2462–2470.

Biographies

Chun-mei Chen received her B.E. degree in Computer Science and Technology and her M.E. degree in Control Engineering from Southwest University of Science and Technology, Mianyang, Sichuan, China, in 2010 and 2000, respectively. She is currently pursuing her Ph.D. degree in Radio Physics major, Institute of Electronic Engineering, China Academy of Engineering Physics. She holds a teacher position with communication engineering working on the teaching and research from 2000 to now. Before her doctoral study, she researches on network security and routing protocols for computer networks. Her research interests include mobile *ad hoc* networks, cognitive radio networks and wireless communications.

He-song Jiang received his Master of Engineering in Signal and Information Processing from the University of Chinese Academy of Sciences in 2009. Now, he is a lecturer of the School of Information Engineering, Southwest University of Science and Technology. His current research interests include cognitive radio and intelligent learning.

Bin Wu received his B.E. degree in Automatic Control from Railway College of Central South University in 1985. He received his M.E. degree in Computer Application and his Ph.D. degree in Control Theory and Control Engineering from the University of Science and Technology Beijing in 1993 and 1999 respectively. Currently, he is the full professor in the School of Information Engineering, Southwest University of Science and Technology, Mianyang Sichuan, China. He is selected as one of the academic and technical leaders in Sichuan. His current research interests are artificial intelligence and application, image processing and machine vision.

Hong Jiang received his B.E. degree in Computer Application from Shenyang Ligong University in 1990. He received his M.E. and Ph.D. degrees in Communication and Information System from the University of Electronic Science and Technology of China in 1999 and 2004, respectively. Currently, he is the full professor in the School of Information Engineering, Southwest University of Science and Technology, Mianyang Sichuan, China. He is selected as one of the academic and technical leaders in Sichuan. His current research interests are in field network technology, wireless communication system and wireless cognitive technology.

Juan Zhang received her B.E. degree in Electronic and Information Engineering of Southwest Normal University in 2005. She received her Ph.D. degree in Signal and Information Processing from Chinese Academy of Sciences in 2012. Currently, she is in the School of Information Engineering, Southwest University of Science and Technology, Mianyang Sichuan, China. Her current research interests are in the cognitive radio and the wireless sensor networks.