# SPECTRUM HANDOFF METHOD BASED ON CHANNEL QUALITY PREDICTION

Kaitian CAO[1], Haiyang LU[2] *

*Spectrum handoff can improve spectrum utilization and alleviate scarcity of spectrum resources. Therefore, spectrum handoff plays an important role in cognitive radio networks (CRNs). In traditional spectrum handoff methods, the throughput of CRNs and the delay of secondary users (SUs) are opposed to a certain extent. To address this issue, aiming at the optimization problem of maximizing the throughput of SUs in CRNs, this paper investigates a method to keep the system's overall benefits when spectrum handoff occurs, and formulates a spectrum handoff model based on delay violation ratio. In this spectrum handoff model, we propose the concept of channel access rate based on the delay violation rate, and derive the access sequence of the target channels. Under the condition of satisfying the delay, the double deep Q-network (DDQN) is used to maximize the throughput of CRNs system through finding the maximum delay violation rate. The simulation results show that under the constraints of the delay, the proposed method can significantly reduce the frequency of spectrum handoffs in CRNs while the system throughput can be achieved at a high level.*

**Keywords**: Cognitive radio networks; delay violation rate; throughput; spectrum handoff

## 1. Introduction

With the rapid development of wireless networks and the emergence of various new applications, the demand for wireless spectrum is growing exponentially. The current fixed spectrum allocation strategy cannot meet the requirements of wireless network growth [1]. To this end, Cognitive Radio (CR) technology, as a new spectrum sharing technology, effectively improves spectrum utilization by allowing secondary users (SU) to access the licensed spectrum unused by primary users (PU) [2], CR has received great attention in recent years. In [1][3][4], cognitive radio networks (CRNs) can use efficient spectrum management and resource allocation technologies to improve spectrum utilization and alleviate the shortage of spectrum resources [5]. In CRNs, spectrum management is composed of four parts: spectrum sensing, spectrum decision, spectrum handoff,

[1] Prof., School of Electrical and Electronic Engineering, Shanghai Institute of Technology, Shanghai, China, e-mail: ktcao@sit.edu.cn
[2] M.Eng., School of Electrical and Electronic Engineering, Shanghai Institute of Technology, Shanghai, China, corresponding author's e-mail: luhaiyang524@163.com

and spectrum sharing, where spectrum handoff refers to the process that the agent is interrupted during the transmission process and resumes the transmission again [6].

In [7], the authors proposed the concept of delay violation rate and backup channel for reducing the system delay. However, the overall throughput of the system is not considered in this model, and the reference to the backup channel will make the spectrum resource even scarcer when the number of SUs is large. A novel spectrum handoff-based sensing-throughput tradeoff scheme in cognitive radio is proposed in [8]. In this scheme, when the SU detects the existence of the primary use, it searches for a new idle channel for transmission to optimize the throughput of the system. It can be seen from Fig. 1 that performing spectrum handoff to idle channels immediately after the SUs are interrupted will cause lots of delay. Literature [9] proposed the aggregation interference control based on decentralized Q-learning in CRNs, and applied Q-learning to spectrum prediction. Nevertheless, for the large Q table, traditional Q-learning is obviously unable to deal with it. In [10, 11], authors proposed a low occupancy channel selection method based on previous channel usage statistics in which only the primary users' activity in the channel is considered, and the SU selects the channel with the highest idle period probability. Moreover, spectrum handoff by multiple users at the same time will result in severe conflicts.

In order to solve the defects in the above literatures, we use double deep Q-network (DDQN) [12] to model spectrum switching in CRNs systems with multiple SUs, and propose a channel quality prediction-based spectrum handoff method (CQP-SH). CQP-SH method considers both the throughput and delay, and maximizes the overall throughput of the CRNs by choosing the optimal maximum delay violation rate. At the same time, the transfer learning [13] method is used to accelerate the learning rate of the newly added SU whose parameters are initialized with those of its nearest nodes. The contributions of this paper are summarized as follows:

(1) The delay quality index and throughput quality index based on delay violation rate are proposed to measure the performance of the channel.

(2) We consider the tradeoff between delay and throughput in CRNs, mathematically model spectrum handoff, and derive the expression of throughput in spectrum handoff.

(3) DDQN can solve the overestimation problem of DQN algorithm and can help to find the maximum delay violation rate; therefore, we investigate DDQN and design the CQP-SH algorithm to optimize spectrum handoff in CRNs.

(4) We perform numerical simulations on the proposed CQP-SH method, and the simulation results show that our method outperforms the methods involved in the comparison in terms of both delay and throughput performance.

In this paper, we mathematically model spectrum handoff and propose the CQP_SH algorithm, and the pseudo-code of the algorithm is given in the paper. By simulating and comparing other methods proposed in the article, our method can greatly reduce the spectrum handoff delay of the system while ensuring the system throughput.

## 2. Spectrum handoff model

### 2.1. Spectrum handoff process

There are usually multiple PUs and SUs in the CRNs system, and each PU has its own dedicated channel. SUs can access channels that are not occupied by the PUs. When the sensing module of the SU in transmission detects that the PU is coming, SU has two options: (1) This SU stays and waits for the current channel to be available again. (2) This SU performs spectrum handoff to access another idle channel.

Due to the randomness of Pus' activities, Sus may encounter many interruptions and experience multiple handoffs among different channels. Fig. 1 shows an example where an SU was interrupted three times during the transmission process and finally completed its transmission. In this example, there are three channels to choose from, the access sequence of SU is channel $Ch_1$、channel $Ch_2$ and channel $Ch_3$. This SU's initial channel is channel $Ch_1$, and the spectrum handoff delay refers to the time elapsed that the data transfer of an SU is interrupted until the transmission start again. The steps of traditional spectrum handoff methods are as follows.

- This SU accesses the channel $Ch_1$ from $t_1$ for transmission. Due to the arrival of the PU, it is interrupted for the first time at $t_3$ and perform spectrum handoff to the channel $Ch_2$.
- When the spectrum handoff is finished, SU accesses the channel $Ch_2$ at $t_4$ for transmission. Due to the arrival of the PU, it is interrupted for the second time at $t_4$ and perform spectrum handoff to the channel $Ch_3$.
- When the spectrum handoff is finished, SU accesses the channel $Ch_3$ at $t_7$ for transmission, and complete the data transmission at $t_{10}$.

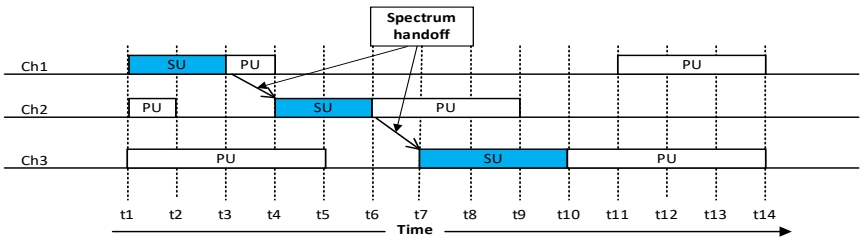


Fig. 1. Spectrum handoff process of secondary user

It can be seen from Fig. 1 that SU frequently performs spectrum handoff creates a large amount of delay. In fact, the duration of PU's existence on channel $Ch_1$ is not too long. The best way for this SU is waiting for PU transmission completion on channel $Ch_1$ and resume transmission, as shown in Fig. 2. The SU took 9 time slots from the start of transmission to the completion of transmission in Fig.1, while in the best mode shown in Fig. 2, the SU has only experienced 8 time slots from the start of transmission to the completion of transmission. Moreover, when multiple SUs in a CRN compete for the right to use spectrum, a large number of handoffs will lead to a greater possibility of contention and collision, resulting in a larger handoffs delay.

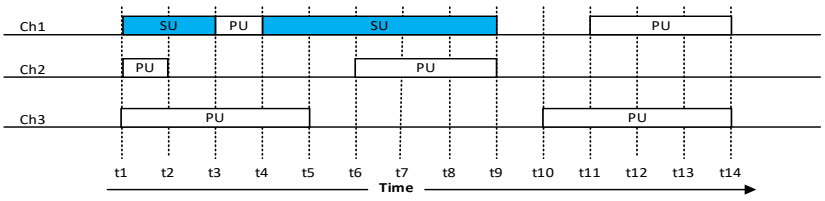


Fig. 2. Optimal spectrum handoff method

### 2.2. Derivation of channel expected delay

As shown in Fig. 3, when $SU_k$ arrives at the queue at $t_k$ it must wait at the end of the queue until all SUs in front of the queue leave.

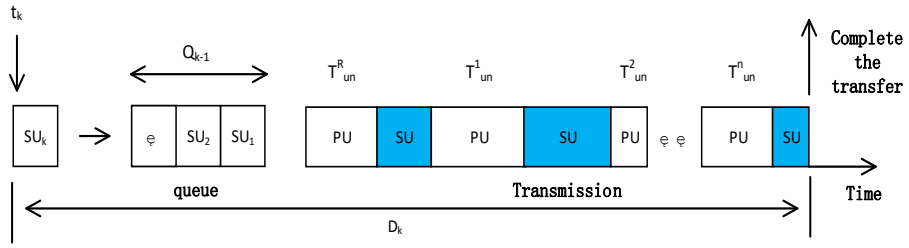


Fig. 3. Calculation of expected delay

We use $Q_k$ to indicate the number of SUs in the queue when $SU_k$ arrive at the queue; $T_{un}^R$ represents the first unavailable time period after reaching the channel; $n_k$ represents the number of times the $SU_k$ was interrupted from the start of the access channel to the end of the transmission, that is, the number of times to resume communication. $T_{un}^i$ indicates the i-th unavailable time period of the channel, that is, the time period during this channel is occupied by the primary user for the i-th time. $D_k$ represents the time from $SU_k$ entering the queue until the transmission is completed, that is, the expected delay. Thus, we can define the expected delay of $SU_k$:

$$E[D_k] = E\left[\sum_{i=1}^{Q_k} S_i + S_i^R + \sum_{i=1}^{n_k} T_{un}^i\right] = Q_k E[S] + E[S_i^R] + E[n_k] E[T_{un}^i] \quad (1)$$

where $E[\cdot]$ is the expectation, $S_i$ is the remaining service time of SUs before in front of queue, and $S_i^R$ indicates the remaining service time of the SU on the channel at the instant of arrival.

In CRNs, since the spectrum environment changes rapidly spectrum handoff requirements that were met at the last moment may not be met at the next moment. Spectrum handoff based on direct delay measurements is impractical as the spectral environment may have changed during the measurement. However, once the SU arrives the queue, the number of SUs $Q_k$, which in the queue can be measured immediately. Therefore, $Q_k$ can be used to estimate the expected delay, which can describe the current state of the channel more effectively than direct delay measurement.

### 2.3. Delay violation rate

$D_{max}$ is the maximum delay allowed by $SU_k$, the delay violation rate of channel $Ch_c$ is $P_v^c$ represents the probability that the expected delay of $SU_k$ over channel $Ch_c$ is greater than its maximum delay $D_{max}$. $P_{max}$ represents the maximum delay violation rate within the allowable range of the $SU_k$. So, the channel availability condition for $SU_k$ is $P_v^c < P_{max}$.

In addition to reflecting the delay requirements, the delay violation rate can also measure the status of the channel. As shown in Fig. 1, traditional spectrum handoff methods perform spectrum handoff immediately after the channel is occupied, then even if the currently occupied time is short, the target channel will be occupied for a long time, and SU will perform the spectrum handoff again. Therefore, we introduce the concept of delay violation rate here. As shown in Fig. 2, with the delay violation rate, the SU can reduce the number of unnecessary spectrum handoff due to intermittent channel availability. In this case, even small spectrum holes can be used for transmission, which avoids waste of resources and does not generate additional handoff overhead.

The expected delay can be calculated by Eq. (1) when the SU arrives at the queue. The queue manager observes the time elapsed between the time each SU arrives at the queue and the completion of the transmission. The delay violation rate can represent the delay performance of the channel for the SU. The new SU, which is about to arrive at the queue, obtains the number of delay violations $N_{max}^w$ by comparing its expected delay over this channel with the previous delay of each SU observed by the queue manager, thereby calculating the delay violation rate of the channel for itself. $N^w$ is the size of the observation window, indicating the number of SUs observed by the queue manager. Therefore, the delay violation rate

expression of the channel can be obtained:

$$P_v = N_{\max}^w / N^w. \tag{2}$$

In the process of calculating the delay violation rate, the observation window size, which is the numbers of SUs, is critical. If the observation window selection is too large, the update cannot be timely; if the observation window is too small, the amount of information is too small to estimate the behavior of the channel because the traffic of SUs and PUs in the channel may change rapidly.

### 2.4. The selection of target channel

This article introduces the concept of the delay violation rate of available channels, and calculates the probability of each channel being selected through the delay quality index and throughput quality index, so as to achieve the purpose of calculating the appropriate $P_{max} \in (minP_v, 1)$ to maximize the throughput. It determines the available channel set $AC = \{P_v^i < P_{max}\}$ after the SU obtains the delay violation rate of each channel, then the delay quality index $Q_D^c$ of channel $Ch_c$ can be expressed as:

$$Q_D^c = \frac{1 - \frac{P_v^c}{P_{\max}}}{\sum_{i \in AC} \left(1 - \frac{P_v^i}{P_{\max}}\right)}. \tag{3}$$

Our goal is to obtain the maximum throughput when the delay requirements are met.Refers to shannon theorem, for any $SU_i$, its throughput $R_i$ on the channel $Ch_c$ can be expressed as:

$$R_i^c = W^c \log_2\left(1 + k \cdot SINR_i^c\right), \tag{4}$$

Where:

$$k = \frac{1.5}{-\ln(5r_b)} \tag{5}$$

$W^c$ is the data bandwidth of channel $Ch_c$; $k$ is a constant determined by the maximum bit error rate $r_b$; $SINR_i^c$ is the Signal to Interference plus Noise Ratio(SINR) of $SU_i$ on channel $Ch_c$. According to the transmit power of the $SU_i$, the SINR value of the $SU_i$ on channel $Ch_c$ can be calculated at the station:

$$SINR_i^c = \frac{G_i^{(s)} P_i}{\sigma^2 + \sum_{j \neq i} G_j^{(s)} P_j}, \tag{6}$$

Where $G_i^{(s)}$ is the channel gain between $SU_i$ and the station; $P_i$ is the transmit power of $SU_i$; $\delta^2$ is Gaussian white noise power; $G_i^{(s)}$ is the channel gain between $SU_i$ and the station. Suppose there are M available channels, Therefore, the

throughput quality index $Q_T^c$ of SU on channel $Ch_c$ is given here:

$$Q_T^c = \frac{R^c}{\sum_{i=1}^{M} R^i}. \tag{7}$$

The delay quality index measures the quality of the target channel in terms of delay for all other available channels. The throughput quality index measures the quality of the target channel in terms of throughput for all other available channels. In order to determine the target channel, the delay and the throughput of the target channel should be considered simultaneously. According to Eq. (3) and Eq. (7), the access probability $P_i^{a,c}$ of $SU_i$ on channel $Ch_c$ can be calculated as:

$$P_i^{a,c} = \frac{(1-\rho)Q_{D,i}^c + \rho Q_{T,i}^c}{\sum_{j=1}^{M}\left[(1-\rho)Q_{D,j}^c + \rho Q_{T,j}^c\right]}, \tag{8}$$

Where $\rho \in [0,1]$ is the throughput bias factor, a discount factor indicating the importance of throughput requirements, the larger the value of $\rho$, the more stringent the throughput requirements and the lower the delay requirements. After obtaining the access probability of each channel, use the channel access probability as the weight to calculate the expected throughput value of the SU:

$$E[R_i] = \sum_{j=1}^{M}\left(R_i^j \cdot P_i^{a,j}\right). \tag{9}$$

This article refers to the M/G/m queuing theory model and ranks the available channels according to their expected throughput. Calculate the access probability of each available channel according to Eq. (8), and arrange the access probabilities in descending order as the order of SU spectrum handoff access.

## 3. Spectrum Handoff Based on DDQN

### 3.1. Double deep Q-networks

Q-learning is a model-free reinforcement learning algorithm that can solve many unsupervised learning problems [14]. When the agent has no knowledge of the environment, it randomly takes actions, and the actions taken according to the current state will get rewards or penalties from the environment. The agent establishes a Q table through the method of continuous trial and error. The Q table stores the Q value of each action-state pair and updates the Q table during each trial and error process. By continuously increasing the number of interactions with the environment, the optimal action set can be finally obtained.

Q-learning finds an optimal strategy in the Markov Decision Processes (MDPs), making it start from the current state, and get the maximum rewards for

each subsequent step. The state of the agent at time $t$ is $s_t$. In this state, an action $a_t$ is selected, and $Q(s_t, a_t)$ is the Q value of this state-action pair. $r_{t+1}$ is the reward obtained by taking action $a_t$ at time $t+1$ in the state $s_t$ at time $t$. Rewards are used to evaluate actions, which can be good or bad. The update method of Q-learning is:

$$Q_t \leftarrow Q_t + \alpha\left(r_{t+1} + \gamma \max Q_{t+1} - Q_t\right), \tag{10}$$

Where $Q_t$ is the Q value at time $t$; $maxQ_{t+1}$ is the maximum Q value at the next time; $\gamma \in (0,1)$ is a discount factor that indicates how far the time affects the rewards; $\alpha$ is the learning rate.

For traditional Q-learning, when the action and state space are discrete and the dimension is not high, the Q table can be used to store the Q value of each state-action pair. However, for the high-dimensional continuous state and action space, a very large Q table is required, which is not feasible, and each time the state is searched in a huge table, it will also occupy huge resources, which is very time-consuming [15]. Therefore, the update of the Q table is converted into a function fitting problem. The Q value obtained in Q-learning is used as a label for deep learning, thereby organically combining reinforcement learning and deep learning to form a Deep Q-Network (DQN).

The neural network used by DQN is trained with a revised Q-learning algorithm, uses gradient descent to update the weights, and uses an experience replay mechanism to eliminate the correlation between the data. First use the deep neural network as the network of Q value, and update the parameters $\theta$ as:

$$Q(s, a, \theta) \approx Q^{\pi}(s, a). \tag{11}$$

Then use the mean square error in the obtained Q value to define the loss function of the objective function:

$$L(\theta) = E\left[\left(r_{t+1} + \gamma \max Q\left(s_{t+1}, a_{t+1}, \theta^{-}\right) - Q(s_t, a_t; \theta)\right)^2\right] \tag{12}$$

Then calculate the gradient of parameter $\theta$ with respect to the loss function as $\nabla_{\theta} L_i(\theta)$ . Then use gradient descent to update the parameters to obtain the optimal Q value.

The purpose of this paper is to calculate the channel access probability according to the system's bias for delay and throughput under the premise of a given requirement $\{D^i_{max}\}$, so as to maximize the overall throughput of the system and determine $\{P^i_{max}\}$ based on this. Since what we seek is the maximum delay violation rate of each SU, and the problem of overestimation of the the traditional DQN will result inaccurate predictions. Therefore, this article uses DDQN to find the optimal solution.

There are two neural network models in DDQN, and there is a certain time difference between the two models, one of which is used to select actions, and the other is used to evaluate the selected actions. The two neural networks can not only

avoid the problem of overestimation, but also eliminate the strong correlation between network weights and loss functions and thus eliminate the oscillations during the training process. DDQN does not directly search for the maximum Q value in each action in the target Q network, but first finds the action corresponding to the maximum Q value in the current Q network, and then uses the action from this faith to calculate the target Q value in the target network.

DDQN randomly initialize all Q values, the weights of Q network, the weights of the target Q network, and empty the experience replay pool D, after which we iterate until the termination state. Use sample $(s_t, a_t, r_t, s_{t+1})$ collected from the experience replay pool to calculate the target Q value during iteration:

$$Q_j = \begin{cases} r_j & , s_{j+1} \\ r_{t+1} + \gamma Q\left(s_{t+1}, \arg\max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta); \theta^-\right) & , s_{j+1}^- \end{cases}. \tag{13}$$

The loss function of DDQN is defined as:

$$L(\theta) = E\left[\left(r_{t+1} + \gamma Q\left(s_{t+1}, \arg\max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta); \theta^-\right) - Q(s_t, a_t; \theta)\right)^2\right]. \tag{14}$$

### 3.2. Channel quality prediction-based spectrum handoff (CQP-SH)

Based on the above introduction, in order to maximize the system throughput under the given maximum delay threshold, assume that the number of SUs in the system is *L* and the number of channels is *Y*, the selection of the maximum delay violation rate can be attributed to the following optimal problem:

$$\begin{gathered} \left\{P_{\max}^i\right\} = \arg\max \frac{1}{L} \sum_{i=1}^{L} E[R_i], \\ s.t.\ P_v^j < P_{\max,i} \end{gathered} \tag{15}$$

where $P_v^j$ is the delay violation rate of channel $Ch_j (j = 1,2,\cdots,Y)$.

Therefore, in DDQN, the action space of SUs is $A = \{+0.01, -0.01\}$. The state can be defined as the maximum delay violation rate of the SUs, and the state space of the system is $S = \{P_{max,1}, P_{max,2}, \cdots, P_{max,L}\}$. Define the reward function as a function on the state space and the current action space, then at time *t*, the reward obtained by $SU_i$ is

$$r_i(s_t, a_t) = \begin{cases} R_i, & P_v^j < P_{\max,i} \\ \Lambda, & P_v^j \geq P_{\max,i} \end{cases}. \tag{16}$$

In Eq. (16), Λ is a constant, and its value is less than the reward value obtained when adopting any learning strategy. Therefore, when $P_v^j \geq P_{max,i}$, the reward obtained by the system is punishment.

The strategy $\varepsilon - greedy$ [12] is adopted in DDQN to select each SU's action (SU's maximum delay violation rate), and update the parameters of the neural network to maximize the reward of $SU_i$, thereby maximizing the overall throughput of the system.

We note that the Q function parameters of two adjacent SUs in the CRNs are similar, the closer the distance, the more similar the parameters. Therefore, this paper uses the method of transfer learning to initialize the newly added SU in the CRNs using the Q function and parameters of the SU closest to it. In this way, for newly joined SUs, there is no need to learn from scratch, which greatly speeds up the learning process and improves the performance of CRNs. The pseudo code of the method CQP-SH using DDQN is given below.

**Algorithm 1:** CQP-SH Algorithm

Initialize replay empirical replay pool to capacity N
Initialize Q value of state-action pair with random weights
**for** epoch = 1, M **do**
Initialize sequence $s_1 = \{P_{max,11}, P_{max,12}, \cdots, P_{max,1L}\}$
**for** t = 1, T **do**
Select a random action $a_t$ with a probability epsilon
**Otherwise** select $a_t = argmaxQ_{max}(s_t, a_t; \theta)$
Execute action $a_t$ in emulator and observe reward $r_t$ and $x_{t+1}$
Set $s_{t+1} = s_t, a_t, x_{t+1}$
Store transition $(s_t, a_t, r_t, s_{t+1})$ in D
Sample random mini batch of transitions $(s_j, a_j, r_j, s_{j+1})$ from D
Set $Q_j$ according to Eq. (13)
Perform a gradient descent in Eq. (14)
**end**
**end**

## 4. Experimental Results

In the simulations, we suppose that the initial value $\varepsilon = 1.0$, and its final value is 0.01. The learning rate decays with each update, and the value of epsilon decay is 500. Two identical neural networks are used for initialization, with one node in the input layer, two nodes in the output layer, and two hidden layers with 128 nodes, and the calculation cost after convolution is 0.66million when only multiplication is considered. The size of the empirical replay pool is 1000, the batch size is 32, and the gamma value is $\gamma = 0.99$.

The throughput bias factor $\rho = 0.3$. Fig. 4 shows the influence of the observation windows on the number of spectrum handoff under the different number of SUs. From Fig. 4, we can see that the larger the observation window size

is, the fewer the number of handoffs is, the lower the delay is. However, the observation window should not be too large, which may cause the update speed to be slow.

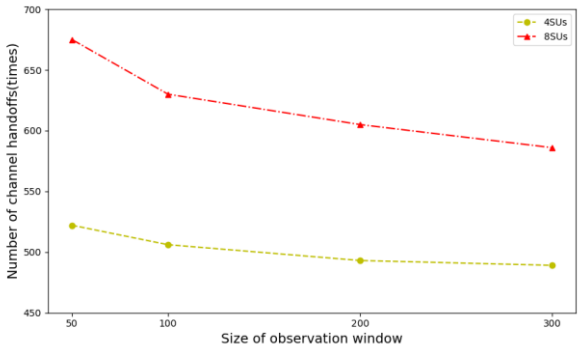


Fig. 4. The influence of observation windows and SUs

The throughput bias factor $\rho = 0.7$, the observation window size $N^w = 200$, Eq. (1) gives the time required by the SU from the start of accessing channel to the completion of the transmission. In our method, the SU will choose whether to perform spectrum handoff according to the specific situation. The transmission time of each SU is fixed. In this paper, the delay is expressed as the number of spectrum handoff. We compared the performance of our proposed CQP-SH method with that of low occupancy [10] and [11], sensing-throughput tradeoff [8] and random selection methods. The random selection method is that SUs select the channel randomly after being interrupted by PUs.

From Fig. 5, We can see that in CRN, the spectrum switching time increases with the number of SUs,, and CQP-SH is significantly more efficient than other methods. In addition, in the random selection method, SUs randomly select idle channels, resulting in a higher spectrum handoff delay as the number of SUs increases. It can be seen from Fig. 8 that our CQP-SH method can achieve greater throughput than other methods.

From Fig. 6 and Fig. 8, when the number of SUs is greater than the number of channels, the increase of the number of SUs does not have a regular effect on throughput since they only consider delay and not throughput. We can see that in Figure 6, the throughput of the random selection method suddenly drops when the number of SUs is 7, and when the number of SUs is 9, it increases again. The throughput of the low occupancy method decreases when the number of SUs is 8, but when the number of SUs is 9, the throughput increases again. In Fig. 8, these two methods also perform poorly in throughput performance, and also show some randomness. Overall, the throughput performance of these two methods remains a dynamic fluctuation in the lower range.

In addition, due to the large value of the throughput bias factor setting, the

requirement for delay is relatively low, and the requirement for throughput is high. It can be seen that when the throughput bias factor's setting is high, our method significantly outperforms other methods in throughput.

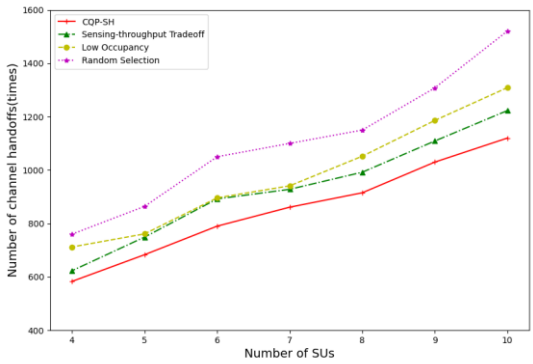


Fig. 5. Number of spectrum handoff vs. number of SUs ($\rho = 0.7$)

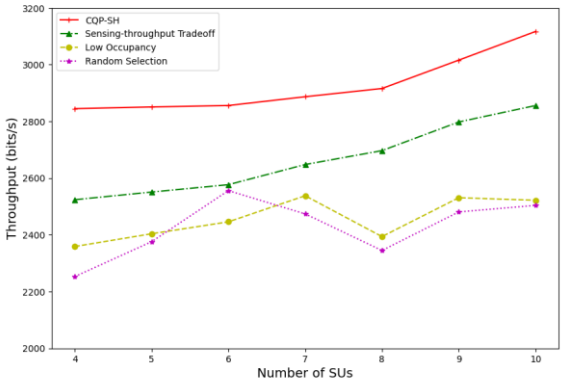


Fig. 6. Throughput vs. number of SUs ($\rho = 0.7$)

We change the value of the throughput bias factor to $\rho = 0.3$, the system has more stringent requirements for delay and relaxed throughput requirements since we adjusted the throughput bias factor down. The simulation results verify this phenomenon, Fig. 7 shows that the time of spectrum handoff of our method is significantly reduced. On the other hand, as shown in Fig. 8, there is no much room for improvement in throughput of low occupancy method and random selection method. Due to the smaller throughput bias factor, we have relatively low throughput requirements; however, our method still outperforms other methods on the whole. In fact, since the introduction of the delay violation rate, the smaller value of the delay violation rate, the better channel's delay is.

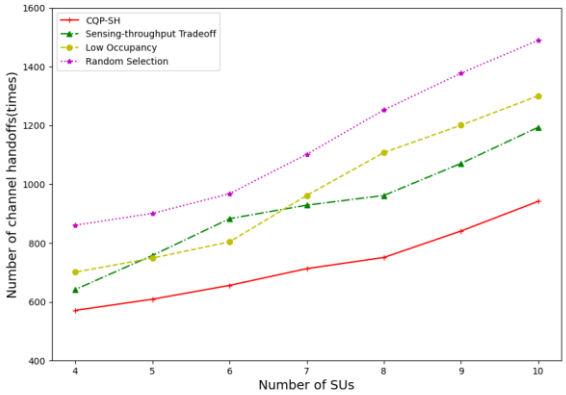


Fig. 7. Number of spectrum handoff vs. number of SUs ($\rho = 0.3$)

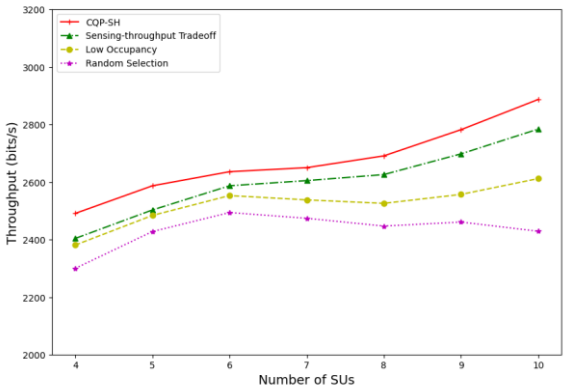


Fig. 8. Throughput vs. number of SUs ($\rho = 0.3$)

The introduction of throughput bias factor allows the system to coordinate throughput and delay, and it can be adjusted to meet different system requirements for complex environments.

## 5. Conclusions

In this paper, DDQN is introduced into the spectrum handoff in CRNs, then the spectrum handoff model based on channel quality prediction is proposed, and finally the CQP-SH spectrum handoff algorithm is designed by introducing the delay violation rate. The simulation results show that our method can guarantee better delay performance while maintaining throughput performance at a relatively high level. In a word, we can set the value of the throughput deviation coefficient to meet the latency and throughput requirements in different environments, so that the system's throughput can be maximized under the premise of meeting the delay

requirements.

## R E F E R E N C E S

[1]. AKYILDIZ I F, LEE W Y, VURAN M C, MOHANTY S. A survey on spectrum management in cognitive radio networks. IEEE Communications Magazine, 2008, 46(4): 40–48.

[2]. WANG Beibei and LIU K J R. Advances in cognitive radio networks: A survey. IEEE Journal of Selected Topics in Signal Processing, 2011, 5(1): 5-23. doi: 10.1109/JSTSP.2010.2093210.

[3]. DING Guoru, JIAO Yutao, WANG Jinlong, et al. Spectrum inference in cognitive radio networks: algorithms and applications. IEEE Communications Surveys & Tutorials, 2018, 20(1): 150-182.

[4]. MA Bin, CHENG Shuangguo, XIE Xianzhong. PRP M/G/m queuing theory spectrum handoff model based on classified secondary users. Journal of Electronics & Information Technology, 2018, 40(8): 1963-1970.

[5]. KOUSHIK A M, HU F, and KUMAR S. Intelligent spectrum management based on transfer actor-critic learning for rateless transmissions in cognitive radio networks. IEEE Transactions on Mobile Computing, 2018, 17(5): 1204-1215.

[6]. TANAB M E and HAMOUDA W. Resource allocation for underlay cognitive radio networks: A Survey. IEEE Communications Surveys & Tutorials, 2017, 19(2): 1249-1276.I. Akyildiz et al., “A Survey on Spectrum Managementin Cognitive Radio Networks,” IEEE Commun. Mag., vol.46, no. 4, Apr. 2008, pp. 40–48.

[7]. A. Lertsinsrubtavee, N. Malouch and S. Fdida, "Controlling Spectrum Handoff with a Delay Requirement in Cognitive Radio Networks," 2012 21st International Conference on Computer Communications and Networks (ICCCN), Munich, 2012, pp. 1-8, doi: 10.1109/ICCCN.2012.6289234.

[8]. X. Liu, F. Li and W. Lu, A novel spectrum handoff-based sensing-throughput tradeoff scheme in cognitive radio, in China Communications, vol. 13, no. 12, pp. 59-68, December 2016, doi 10.1109CC.2016.7897555.

[9]. A. Galindo-Serrano and L. Giupponi, "Distributed Q-Learning for Aggregated Interference Control in Cognitive Radio Networks," in IEEE Transactions on Vehicular Technology, vol. 59, no. 4, pp. 1823-1834, May 2010, doi: 10.1109/TVT.2010.2043124.

[10]. L. Yang, L. Cao, and H. Zheng, “Proactive channel access in dynamic spectrum networks,” Physical Communication, vol. 1, no. 2, pp. 103–111, 2008.

[11]. WANG Beibei and LIU K J R. Advances in cognitive radio networks: A survey. IEEE Journal of Selected Topics in Signal Processing, 2011, 5(1): 5-23. doi: 10.1109/JSTSP.2010.2093210.

[12]. Van Hasselt H, Guez A, Silver D. Deep Reinforcement Learning with Double Q-learning. Computer Science, 2015.

[13]. PAN S J and YANG Qiang. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(10): 1345-1359.

[14]. A. Brim, "Deep Reinforcement Learning Pairs Trading with a Double Deep Q-Network," 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2020, pp. 0222-0227, doi: 10.1109/CCWC47524.2020.9031159.

[15]. MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518(7540): 529-533.