Distributed Channel Selection in Opportunistic Spectrum Access: A Learning-Based Approach

Name: Xu Wang, Student Number: 5100309397

Abstract

Opportunistic spectrum access (OSA) has been regarded as the most promising approach to solve the paradox between spectrum scarcity and waste. And we investigate the practical OSA model, where secondary users make their channel selections in a completely distributed fashion. We formulate a game structure to help the secondary users make decisions. Then we prove the proposed game an exact potential game, so that the convergence of the model is guaranteed. However the proposed problem is an NP hard problem. Optimal solution is hard to find without large amount of computation. So we proposed a learning algorithm, which only needs the local information that secondary users can access but has a fast convergence speed to Nash Equilibria(NE) points, to solve the NP hard problem formed in the model. The convergence of the proposed algorithm is proved. And we use some simulation results to show the fast convergence property. Finally we point out some possible ways to extend the research.

Keywords: Opportunistic spectrum access, Learning algorithm, Fast convergence

1 Introduction

There is a common belief that we are running out usable radio frequencies. Opportunistic spectrum access, which is based on cognitive radio is always regarded as one of the most promising solution to the spectrum scarcity problem. In OSA systems, there are two types of users. One is the primary user which is the licensed owner of the spectrum, and the other is the cognitive user(secondary user) which is allowed to transmit in the licensed spectrum at a particular time and location when and where the primary users are not active.[1] Usually, the channel selection in OSA systems is centralized, which means that a secondary user is required to have a global knowledge about the system and other secondary users. To make the most of spectrum resources and make the system more practical, distributed channel selection , in which secondary users only use their local information to estimate the quality of the available channels, should be applied. Learning algorithm is the most useful tool to solve such a problem.

In most of current works on learning, they focus on the convergence of the algorithm but forget the convergence speed is important as well. Nowadays, the environment is time variant, but most of systems are under a static framework. These two factors will lead to the problem that not until the system reaches the stationary point, the environment changes. So increasing the convergence speed in a static model is desirable. That is what we are focusing on. By extracting the useful information from the limited resources that a secondary user can access, we are able to develop an algorithm which increases the convergence speed by 30%.

The whole report is organized as follows. Section 2 introduces some previous works related to learning approaches. Section 3 describes the system model and how the problem is formulated and also the proof of the convergence of the problem. In section 4, we briefly introduces the existing algorithm used to solve the problem and point out some drawbacks of it. Section 5 mainly describes the algorithm we proposed and proves its convergence. In section 6, some simulation results are displayed to show the fast convergence property. Section 7 draws a conclusion and Section 8 raises some upcoming challenges derived from this problem.

2 Some related work

2.1 Stochastic learning automata based algorithm

In [2], the authors make the channel selection decentralized by a stochastic learning automata(SLA) based algorithm. In their scheme the secondary users can make estimation about how crowded a channel is and automatically adjust its knowledge about the channels using only the reward they get from accessing the channel. Finally the solution will converge to a pure strategy, which means in a time slot, the secondary users are able to select a channel with 100 percent probability. And it is proved that the strategy is a NE point.

2.2 Regret based algorithm

In [3], the authors estimate channel contention for a simple CSMA channel sharing scheme. They use a game theoretic framework to highlight issues of competition among multiple radios. In the resource allocation part, the authors proposed a regret tracking algorithm to estimate the status of each channel. But the authors fail to prove its convergence to the NE point, they only find it converge to a correlated equilibria. And this algorithm is completely distributed, each secondary user only uses its local information to make judgement.

2.3 Spatial adaptive play based algorithm

In [4], the authors consider the resource allocation problem from two level. The first level is a central algorithm, which means there is a central station to collect information from all the secondary users and allocate the channel resources to achieve the global optimal. Then they make the problem more challenging to remove the central station so that secondary users have to learn about channel information, including the quality and contention level, on their own through the interaction with the system. The authors proposed a learning algorithm called concurrent spatial adaptive play(C-SAP) based on the SAP algorithm to solve such a problem. It will help the system achieve global optimal throughput and also maintain a tolerable level of unfairness.

3 System Model

3.1 Model Description

We use a similar system model with that in [2]. In our model, there are N secondary users and M liscenced channels($0 < M < N$),. Each channel has a maximum transmission rate R_m, $0 < m < M$. Here we assume that the same channel has the same transmission rate for all users, although different users may experience different transmission environment, like channel fading and transmission distance. Moreover, it is assumed that the primary users use the licensed channels in a slotted fashion and their activities are independent from channel to channel and from slot to slot. Under such an assumption, we define θ_m as the probability of channel m to be idle and the value of θ_m is time invariant. To make the model more practical and energy efficient, we assume that secondary users do not exchange information with each other and the channel availability statistics θ_m is unknown.

When it comes to the transmission procedure, we consider a slotted structure. And each time slot has the structure as illustrated in Figure 1. The first step is channel selection and channel sensing. Channel selection in this model is completely distributed and we assume that the channel sensing is perfect in this case. Suppose the total length of a time slot is T, the channel selection and sensing time is set to be a fixed value T_s. The remaining length of this time slot is represented as T_e. The second step is channel contention. If a channel is sensed to be idle in the first step, then the secondary user will attempt to use this channel for its own

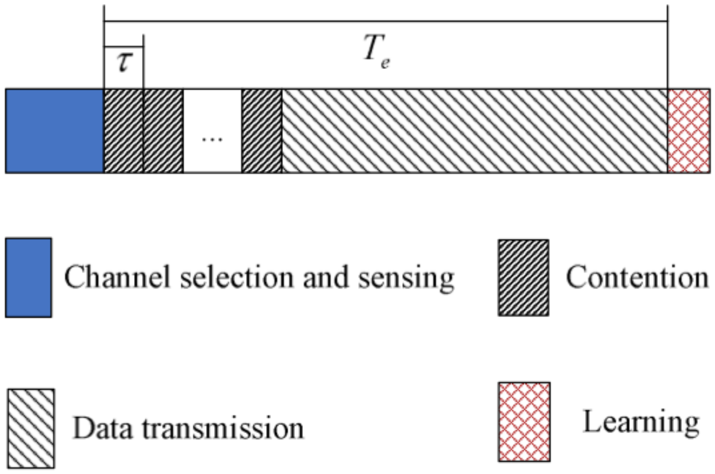

Figure 1: Illustration of a single time slot

transmission. To avoid collision with other secondary users, we use a Carrier Sensing Multiple Access(CSMA) scheme here. T_e is divided into mini time slots with length $\tau(\tau << T_e)$. Here we consider two types of CSMA.

- Type (I)
  In each mini time slot, each secondary user contends for the channel it selects with a fixed probability P_a.
- Type (II)
  Each contending secondary user generates a random integer w_n, which follows uniform distribution on $[1, W_{max}]$. W_{max} here is supposed to be comparable with N. Then the secondary user will contends the channel on the w_nth mini time slot.

A channel contention of a secondary user is said to be successful if no other secondary user contends during the same mini slot. The successful secondary is allowed to transmit its data for the remaining time of this slot. It is clear that Type(I) is suitable for the condition that N is not much larger than M, because it encourages secondary users to contend for the channel as soon as possible. Type(II) is suitable for the situation that N is much larger than M, because it separates secondary users uniformly in the interval $[1, W_{max}]$.

The final step it to extract information from the access experience in this time slot, or in other word learning from the process. Although this step will cost a small portion of this time slot, we consider it to be ignorable.

3.2 Problem Formulation

We first analyze the problem in a single time slot. Let a_n denote the channel selection of user n and s_m denote the number of secondary users contending for channel m. Then we can have the random reward received by secondary user n:

$$r_n = [(T_e - N_c(s_m)\tau)/T_e]\beta_n(s_m)I_m R_m \tag{1}$$

In (1), $N_c(s_m)$ is the number of mini time slots it takes for a successful channel contention with the condition that there are s_m secondary users contending for this channel. $\beta(s_m)$ is an indicate function to represent if user n is able to successfully contend for channel m. I_m indicates whether channel m is idle in this time slot.

If we choose Type(I) as a CSMA scheme, then we have $N_c s_m$ as a geometric random variable with the probability mass function (2):

$$Pr\{N_c(s_m) = i\} = p_s(1 - p_s)^{i-1} \quad i > 0 \tag{2}$$

where $p_s = s_m p_a (1 - p_a)^{s_m - 1}$ represents the overall successful channel contention probability in a mini time slot.

When Type(II) CSMA scheme is adopted, then we can get the distribution of $N_c(s_m)$:

$$Pr\{N_c(s_m) = i\} = \frac{1}{W_{max}}(1 - \frac{1}{W_{max}})^{(i+1)s_m - 1} \quad i > 0 \tag{3}$$

Based on the analysis above, we can have the expected reward achieved by secondary user n in a time slot j:

$$\bar{r}_n(j) = \theta_m f\left(s_m\right) R_m \tag{4}$$

where $f(s_m)$ is a function of s_m representing the expected throughput loss with s_m secondary users contending for channel m:

$$f\left(s_m\right) = \frac{E[T_e - N_c(s_m)]/T_e}{s_m} \tag{5}$$

Actually, our following analysis is not quite related to the expression of $f(s_m)$, thus we can use $f(s_m)$ instead of the specified expression of it. As we know, when a secondary user fails contending for a channel, it will still waste some energy and more or less affects the transmission of the successful one. Thus we give these users a small penalty here. This is intended to improve the convergence speed which will be mentioned in the following sections. To make the penalty more accurate, we define the number of mini time slots it takes for a secondary user n to contend for a channel m as $N_{t_{nm}}(s_m)$. Under our assumption, each user knows $N_{t_{nm}}$ so that it can be used in the distributed algorithm. Also the reward defined in (1) is also the information we can exploit. Then we can define the revenue of user n as

$$\xi_n = \begin{cases} \frac{r_n N_{t_{na_n}}(s_{a_n})}{R_{max}}, & r_n \neq 0 \\ -\frac{1}{N_{t_{na_n}}(s_{a_n})}, & r_n = 0 \end{cases} \tag{6}$$

where R_{max} is the maximum transmission rate of all the m channels and is a fixed value to normalize the reward of each user. We use a_n in (6) instead of m to represent the channel selection of user n, because a secondary user only receives the information from the channel that it chooses in current time slot.

It is obvious to see that when secondary user n contends for channel m with $s_m - 1$ other users, the probability that user n wins the contention, i.e. $r_n > 0$, is $\frac{1}{s_m}$. Then we can calculate the expectation of revenue ξ_n as (7):

$$\bar{\xi}_n = E\left[\xi_n\right] = \frac{\theta_m f(s_m) R_m N_c(s_m)}{R_{max}} \frac{1}{s_m} - (1 - \frac{1}{s_m})L \tag{7}$$

where L denotes the expectation of the number of mini times slots that a secondary user has to wait to contend for the channel. When we adopt Type(I) CSMA scheme, $L = \frac{1}{P_a}$. If we use Type(II) CSMA scheme, $L = \frac{W_{max}}{2}$.

Then we can define the total revenue of the system as (8):

$$U_s(a) = \sum_{n=1}^{N} \bar{\xi}_n \tag{8}$$

where $a = (a_1, a_2, \ldots, a_N)$ is the channel selection profile for the secondary users. Then the system-centric objective is to find the optimal channel selection profile a_{opt} such that the system throughput is maximized:

$$a_{opt} = \arg\max U_s(a) \tag{9}$$

Solving the equation in (9) is challenging with no central controller here to exchange information here, so a distributed approach with learning ability is desirable. This model can be modeled as a non-cooperating game to get a suboptimal solution.

The game is denoted by $G = \left[\mathrm{N}, \{A_n\}_{n\in\mathrm{N}}, \{u_n\}_{n\in\mathrm{N}}\right]$, where N is the set of players, $A_n = \{1, \ldots, M\}$ is the set of available actions for player n and u_n is the utility function for player n. The utility function here can be defined as the expected achievable throughput of player n:

$$u_n(a_n, a_{-n}) = E\left[\xi_n|(a_n, a_{-n})\right] \tag{10}$$

in (10), a_{-n} represents the channel selection profile of all players excluding n.

3.3 Convergence of the problem

The key idea to prove the convergence is to prove that the game we proposed here is an exact potential game. And according to the good properties of exact potential game, the convergence to NE point will follow.

We can see from (7) that the expectation of ξ is the function of s_m and s_m is the only variable in this equation, which means the expectation of ξ is only related to the number of users who contends for channel m, instead of which users are contending for the channel. This allows us to adopt the potential function described in [6], i.e. we can define the potential function here as (11):

$$\Phi(a_n, a_{-n}) = \sum_{m=1}^{M}\sum_{k=1}^{s_m} \varphi_m(k) \tag{11}$$

where $\varphi_m(k) = \frac{\theta_m f(s_m) R_m N_c(s_m)}{R_{max}} \frac{1}{s_m} - (1 - \frac{1}{s_m})L$.

Theorem 1. *The proposed non-cooperating game G, is an exact potential game with the potential function defined by (11).*

Proof. When a single secondary user n changes its selection unilaterally from a_n to $\tilde{a}_n$, then the change on user n's utility function is given by:

$$u_n(\tilde{a}_n, a_{-n}) - u_n(a_n, a_{-n}) = \varphi_{\tilde{a}_n}(s_{\tilde{a}_n} + 1) - \varphi_{a_n}(s_{a_n})(s_{a_n}) \tag{12}$$

Then we can calculate the change on potential function made by the unilateral change of user n:

$$\begin{aligned}\Phi(\tilde{a}_n, a_{-n}) - \Phi(a_n, a_{-n}) &= \Big(\sum_{k=1}^{s_{\tilde{a}_n}+1} \varphi_{\tilde{a}_n}(k) + \sum_{k=1}^{s_{a_n}-1} \varphi_{a_n}(k)\Big) - \Big(\sum_{k=1}^{s_{\tilde{a}_n}} \varphi_{\tilde{a}_n}(k) + \sum_{k=1}^{s_{a_n}} \varphi_{a_n}(k)\Big) \\ &= \varphi_{\tilde{a}_n}(s_{\tilde{a}_n} + 1) - \varphi_{a_n}(s_{a_n})\end{aligned} \tag{13}$$

Then from (12) and (13), we can see that:

$$\Phi(\tilde{a}_n, a_{-n}) - \Phi(a_n, a_{-n}) = u(\tilde{a}_n, a_{-n}) - u(a_n, a_{-n}) \tag{14}$$

We can see from (14) that the change of a single user's utility caused by his unilateral change is equal to the change of the potential function.Then we can say that **Theorem** 1 follows. □

With **Theorem** 1, we can say that there is at least one NE point in the proposed game model and the system will converge to a NE point in finite time durations.

4 The SLA based algorithm

To achieve the NE point, the authors in [2] proposed an effective SLA based algorithm. It is completely distributed and is able to converge to the NE point of the game G in a limited time duration. They use the reward that the secondary users can get in each time slot as the learning material to update the channel selection profile of each secondary user and this the selection of each secondary user is proved to converge to a pure strategy NE point.

However the algorithm that the authors proposed have the following drawbacks:

- This algorithm does not exploit all the information that a secondary user can access, actually they use only the reward that the secondary users get to update the channel selection profile.
- This algorithm suffers a relatively slow convergence speed, because users who have no reward will have to keep unchanged in the current time slot instead of modifying the selection profile.

To solve these problems, I introduce the revised SLA based algorithm, which makes the most of information that a secondary user can get and have a faster convergence speed.

5 The Revised SLA based algorithm

The main idea comes from queuing. Intuitively, in CSMA when a user comes to the channel very early but still cannot get reward, it is reasonable to guess that this channel is very crowded. If a user comes to the channel late and finally gets the reward, then it is highly possible that users contending for this channel is not in a large number. In other word, when updating the selection profile, we can take the number of mini time slots it takes when a secondary user contends for the channel, i.e. $N_{t_{nm}}$ into consideration. This value will help us upgrade the convergence speed. With the revenue defined in (6), we can easily apply this idea into an algorithm.

5.1 Algorithm Description

We first define the channel selection profile of each user n as $\mathbf{p}_n = \{p_{n1}, p_{n2}, \ldots, p_{nM}\}$, where p_{nm} represents the probability of secondary user n selecting the channel m. And the whole channel selection profile can be denoted as $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$.

We can use the revenue of user n ξ_n as a learning factor, our algorithm is described in **Algorithm** 1. In our algorithm, the reward r_n and the number of time slots it takes for a secondary user to contend for the channel $N_{t_{nm}}$ as two reinforcement signal. If the feedback is a positive reward, according to our definition in (6), the learning factor will be proportional to the product of r_n and $N_{t_{nm}}$. Here we can not use N_c and N_t interchangeably, because the positive r_n implies that for this secondary user n, $N_{t_{nm}} = N_c(s_m)$.

To analyze the reasonability of the learning factor ξ_n, we replace r_n with (1), it is easy to find that:

$$\xi_n = \frac{[(T_e - N_c(s_{a_n})\tau)/T_e] R_{a_n} N_c(s_{a_n})}{R_{max}} \tag{16}$$

We take the derivative of ξ_n about $N_c(s_{a_n})$, then we have:

$$\frac{d\xi_n}{dN_c(s_{a_n})} = \frac{R_{a_n}\left[T_e - 2\tau N_c(s_{a_n})\right]}{T_e R_{\max}} \tag{17}$$

In (17), we can see that with the assumption $T_e >> \tau$, we can conclude that $\frac{d\xi_n}{dN_c(s_{a_n})}$ is larger than 0, thus the learning factor will increase with $N_c(s_{a_n})$. This is desirable, because the later

Algorithm 1 Revised SLA based algorithm

1. Initially, set $j = 0$ and the initial channel selection probability vector $p_{nm} = \frac{1}{M}, \forall n \in \mathrm{N}$
2. At the beginning of the jth slot, each secondary user n selects a channel $a_n(j)$ according to its current channel selection probability vector $\mathbf{p_n}(j)$.
3. In each slot, the secondary users perform channel sensing and channel contention. At the end of the jth slot, each secondary user n receives the random reward $r_n(j)$ represented by (1), also it will record $N_t(j)$ for learning.
4. All the secondary users update their channel selection profile according to the following rule:
$$\begin{aligned} p_{nm}(j+1) &= p_{nm}(j) + b\xi_n(j)\left(1 - p_{nm}(j)\right), \quad m = a_n(j) \\ p_{nm}(j+1) &= p_{nm}(j) - b\xi_n(j) p_{nm}(j), \quad m \neq a_n(j) \end{aligned} \tag{15}$$
where b is a small step size to modify the convergence speed of the algorithm.
5. If $\forall n \in \mathrm{N}$, there is a possibility component $p_n(j)$ is larger than a fixed number, which is approaching 1, e.g. 0.99, stop; otherwise go to step 2.

---

a secondary user successfully contends for a channel, the more probable that this channel has less competitors.

However, when the reward a secondary user receives is 0, in other word, it fails in the contention for the channel. it will get a negative penalty, and the absolute value of the factor is reversely proportional to $N_{t_{na_m}}$. This is reasonable, because when a secondary user contends a channel early but still fails in the competition, it is then highly possible that this channel is crowded, and the secondary user must decrease the possibility that it chooses this channel in future time slots.

5.2 Convergence of the algorithm

In [2], the authors proved that with a small step size the SLA based algorithm will converge to the NE point. After making some changes to the algorithm, we have to reconsider the convergence. Fortunately, by using the similar structure with the authors of [2] and some transform of the parameters in the model, we are able to prove the convergence of this fast-convergence algorithm.

The key idea of proving the convergence is to characterize the long term property of the selection profile $\{\mathbf{P}(j)\}$ with an ordinary differential equation and to relate the NE point of our game model to the stationary point of the ordinary differential equation. Then we prove **Theorem 2** when our algorithm is applied. This gives us a sufficient condition to achieve NE points for the learning algorithm and we can prove that our algorithm can satisfy the sufficient condition.

In [5], it is proved that with a sufficiently small step size b, the sequence $\{\mathbf{P}(j)\}$ will converge to $\mathbf{P}^*$, which is the solution of the following ordinary differential equation:

$$\frac{d\mathbf{P}}{dt} = F(\mathbf{P}), \mathbf{P}_0 = \mathbf{P}(0) \tag{18}$$

where $\mathbf{P}_0$ is the initial channel selection probability matrix and $F(\mathbf{P})$ is the conditional expected function defined as:

$$F(\mathbf{P}) = E[T(\mathbf{P}(j))|\mathbf{P}(j)] \tag{19}$$

where $T(\mathbf{P}(j))$ represents the probability updating rule specified by (15).

Also, according to **Theorem 3.2** in [5], we can see that:

- All the stable stationary points of (18) are the NE points of the game G.
- All the NE points of game G are the stable stationary points of (18).

Let $h_{nm}(\mathbf{P})$ denote the expected reward function of player n if $a_n = m$ and other users employ mixed strategy $\mathbf{p}_k$. $h_{nm}(\mathbf{P})$ can be represented by (20).

$$h_{nm}(\mathbf{P}) = \sum_{a_k, k \neq n} u_n(a_1, \ldots, a_{n-1}, m, a_{n+1}, \ldots, a_N) \prod_{k \neq n} p_{ka_k} \tag{20}$$

where u_n is defined in (10).

Theorem 2. *Suppose that there is a non-negative function $H(\boldsymbol{P}) : \boldsymbol{P} \to R$ for some positive constant c such that:*

$$H(m_1, \boldsymbol{P}_{-n}) - H(m_2, \boldsymbol{P}_{-n}) = c(h_{nm_1}(\mathbf{P}) - h_{nm_2}(\mathbf{P})), \forall n, m_1, m_2 \tag{21}$$

where $H(m, \boldsymbol{P}_{-n})$ is the value if H on the condition that $a_n = m$, and $h_{nk}(\boldsymbol{P})$ is specified by (20). Then we come to the conclusion that the revised SLA algorithm converges to a pure strategy NE point.

Proof. First, we consider the long term probability behavior of a single user instead of the complete channel selection profile, then the ordinary differential equation (18) can be written as:

$$\frac{dp_{nm}}{dt} = qF_{nm}(\mathbf{P}), \forall n \in \mathrm{N}, 1 \leq m \leq M \tag{22}$$

where q is a constant factor. Then applying $F_{nm}(\mathbf{P})$ to (22), we can have (23):

$$\frac{dp_{nm}}{dt} = q\left(p_{nm}(1 - p_{nm})E\left[\xi_n|(m, \mathbf{P}_{-n})\right] - \sum_{k=1, k \neq m}^{M} p_{nk}(p_{nm})E\left[\xi_n|(k, \mathbf{P}_{-n})\right]\right) \tag{23}$$

By extracting p_{nm}, and expressing $E\left[\xi_n|(k, \mathbf{P}_{-n})\right]$ in $h_{nm}(\mathbf{P})$, we have:

$$\frac{dp_{nm}}{dt} = q\left(\sum_{k=1}^{M} p_{nk}\left[h_{nm}(\mathbf{P}) - h_{nk}(\mathbf{P})\right]\right) \tag{24}$$

Because we have $H(\mathbf{P}) = \sum_{m=1}^{M} p_{nm} H(m, \mathbf{P}_{-n})$, then the variation of $H(\mathbf{P})$ can be denoted as:

$$\frac{\partial H(\mathbf{P})}{\partial p_{nm}} = H(m, \mathbf{P}_{-n}) \tag{25}$$

Consider the behavior of $H(\mathbf{P})$:

$$\frac{dH(\mathbf{P})}{dt} = \sum_{n,m} \frac{\partial H(\mathbf{P})}{\partial p_{nm}} \frac{dp_{nm}}{dt} \tag{26}$$

Applying (24)(25) to (26), we have

$$\frac{dH(\mathbf{P})}{dt} = \frac{q}{2} \sum_{n,m,k} \left[H(m, \mathbf{P}_{-n}) - H(k, \mathbf{P}_{-n})\right] p_{nm} p_{nk} \left[h_{nm}(\mathbf{P}) - h_{nk}(\mathbf{P})\right] \tag{27}$$

Then we apply (21) to (27), we have:

$$\frac{dH(\mathbf{P})}{dt} = \frac{qc}{2} \sum_{n,m,k} p_{nm} p_{nk} [h_{nm}(\mathbf{P}) - h_{nk}(\mathbf{P})]^2 \geq 0 \tag{28}$$

If we have $\frac{dH(\mathbf{P})}{dt} = 0$, and as we know $p_{nk} \geq 0, \forall 1 \leq k \leq M$, then we can have $[h_{nm}(\mathbf{P}) - h_{nk}(\mathbf{P})] = 0$. Then we can have $F_{nm}(\mathbf{P}) = 0$. Thus we can come to the conclusion that $\mathbf{P}$ is the stationary point of the ordinary differential equation (18). In other words, the sequence $\{\mathbf{P}(j)\}$ converges to a stationary point of (18). Therefore, Theorem 2 follows. □

If we take $H(\mathbf{P}) = E[\Phi(\mathbf{P})]$, then we can have

$$H(m, P_{-n}) = \sum_{a_k, k \neq n} \Phi(a_1, \ldots a_{n-1}, m, a_{n+1}, \ldots a_N) \prod_{k \neq n} p_{ka_k} \tag{29}$$

Considering the definition of $h_{nm}(\mathbf{P})$ in(20) and the relation mentioned in (14), it is clear that:

$$H(m_1, \mathbf{P}_{-n}) - H(m_2, \mathbf{P}_{-n}) = h_{nm_1}(\mathbf{P}) - h_{nm_2}(\mathbf{P}) \tag{30}$$

By **Theorem** 2, we can say that the proposed revised SLA based channel selection algorithm converges to a pure NE point of the game.

6 Simulation Results

We do some simulations on matlab to testify the fast convergence property of our algorithm and compare it with the existing algorithm. Here we set the the total time slot length $T = 100 \times 10^{-3}s$; moreover, to meet the sensing requirement, the sensing length in a slot is set to $T_s = 5 \times 10^{-3}s$. As a result, the time length after channel sensing in a slot is $T_e = 95 \times 10^{-3}s$. The mini-slot length is set to $\tau = 2 \times 10^{-3}s$ and the access probability is set to $p_a = 0.3$(we only consider Type I CSMA scheme here in order for comparison). In addition, the step size of the learning algorithm is set to $b = 0.15$. Then we set the transmission rate of these three channels as follows: $R_1 = 2$, $R_2 = 1.5$, $R_3 = 1$. Finally, we set the idle probability of each channel: $\theta_1 = 0.6$, $\theta_1 = 0.7$, $\theta_1 = 0.6$.

As our main purpose is to improve the convergence speed, we only make some comparison on the convergence speed. There is no need to worry about the system performance loss as this system model can not guarantee the uniqueness of NE points. As long as it converges, it is a suboptimal solution. There is only one optimal NE point, but neither of the two algorithms to be compared can guarantee the convergence to that point. Even in [2], the throughput gap between learning algorithm and random selection(which represents the worst case) is trivial, no larger than 5%. Therefore it is reasonable to focus on the improvement on convergence speed.

To compare with the SLA based algorithm in [2], we pick the long term behavior of the channel selection profile of a single user as an index of convergence. Figure 2 and Figure 3 shows the change of the channel selection probability of secondary user 1 when adopting SLA based algorithm and revised SLA based algorithm respectively. We can see from Figure 2 and Figure 3 that both algorithms help converge to a pure strategy solution. But careful observation can reveal that the the revised SLA based learning algorithm converges a little bit earlier than SLA based algorithm.

To avoid contingency, we make 200 tests to capture the number of iterations needed for convergence of the system. The result is shown in Figure 4. The red line in Figure 4 shows the numbers of iterations needed by the SLA based learning algorithm to converge to the NE point in each test. And the blue lines shows the same meaning for the revised SLA based algorithm. It is obvious that red lines are above blue lines in most situations, which means that the SLA based algorithm needs more time to converge than the revised one. To be more accurate, we calculate the mean value of time iterations needed for convergence. For the existing SLA based

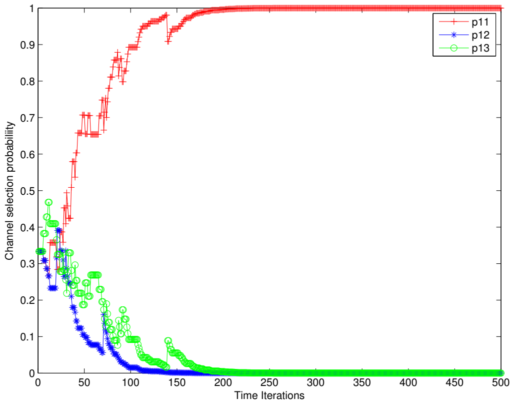

Figure 2: SLA based algorithm

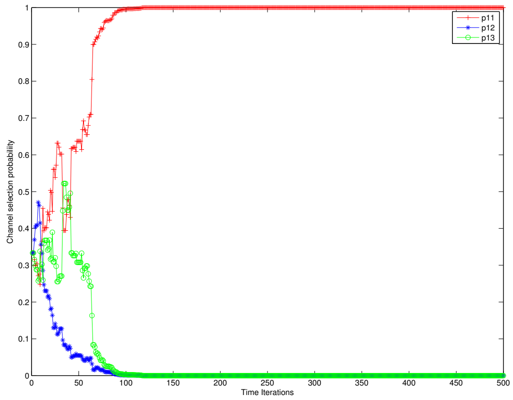

Figure 3: Revised SLA based algorithm

algorithm, the mean value is around 170. And for the revised one, the mean value is around 120. That means the upgrade on convergence speed is more than 30%.

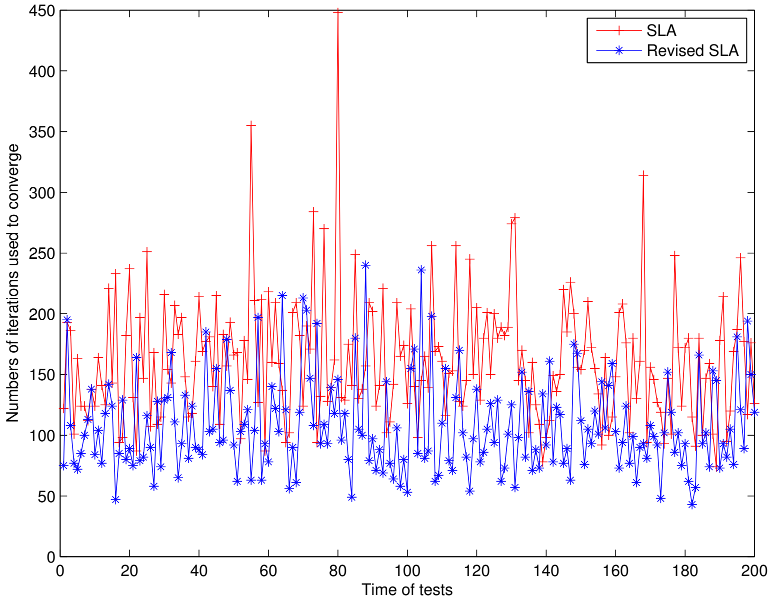

Figure 4: The convergence speed comparison

7 Conclusion

We studied the problem of distributed channel selection under the structure of OSA. We then used a game structure to formulate the channel selection problem. Then by proving it an exact potential game, we clarified the existence of NE points of the game, but the uniqueness of the game model cannot be guaranteed. Then to solve the NP hard problem formulated by game model, we developed a learning algorithm, which is based on stochastic learning automata, to reach the NE point of the game. Then we proved the proposed algorithm is able to converge to the NE points of the game. Compared with existing algorithm, the proposed algorithm utilizes the same local information, but enjoys a better convergence speed. This property can be seen from several simulation results.

8 Future work

Actually, the focus of this report is basically on the contention part of channel selection, so we make the assumption that the channel sensing is perfect. But in practical, the 100% correctness cannot be guaranteed. Thus for the next step, considering the sensing error in the model may make it more practical. In addition, lack of theoretical analysis on the performance of the proposed algorithm is also a weakness here. Furthermore, in our model, all the users are viewed equally, which means each two secondary users are equivalent. This is not practical either, because each secondary user in the real world will suffer from different transmission condition. As a result, even the same channel may be different in different users' eyes. Therefore, it is desirable to extend the existing model to a user specific one, so that all the practical situations can be modeled. Some other researchers may have noticed such problems, and have already made some progress, like [4] mentioned the spacial reuse to characterize the difference among secondary users, and [7] introduced the movement of secondary users. So, we still have a lot to do on this topic.

References

[1] Xu, Y., A. Anpalagan, et al."Decision-Theoretic Distributed Channel Selection for Opportunistic Spectrum Access: Strategies, Challenges and Solutions." , 2013

[2] Xu, Y., J. Wang, et al. "Opportunistic Spectrum Access in Unknown Dynamic Environment: A Game-Theoretic Stochastic Learning Solution." , IEEE transaction on Wireless Communication, 11(4): 1380-1391, 2012

[3] Maskery, M., V. Krishnamurthy, et al. "Decentralized dynamic spectrum access for cognitive radios: cooperative design of a non-cooperative game." , IEEE Transactions on Communications, 57(2): 459-469, 2009

[4] Xu, Y., J. Wang, et al. "Opportunistic spectrum access in cognitive radio networks: global optimization using local interaction games." , IEEE Journal of Selected Topics in Signal Processing, 6(2): 180-194, 2012.

[5] P. Sastry, V. Phansalkar, and M. Thathachar, "Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information," IEEE Trans.on Systems, Man and Cybernetics, 24(5), 769-777, 1994.

[6] B. Vcking and R. Aachen, "Congestion games: optimization in competition," in Proc. 2006 Algorithms and Complexity in Durham Workshop, pp. 9-20.

[7] Chen, X. and J. Huang "Distributed spectrum access with spatial reuse." , IEEE Journal on Selected Areas in Communications, 31(3): 593-603, 2013