Decentralized Spatial Spectrum Access

Bangyi Zhu, Xu Chen, and Jianwei Huang

***Abstract*—In this paper, we study the distributed spectrum sharing problem with spatial reuse and without explicit message passing. We propose tow novel threshold-based decentralized spatial spectrum access algorithms, which do not require information exchange among secondary users and channel switching at the equilibrium state. Moreover, we show that the proposed algorithms can converge to either an approximate Nash equilibrium or an approximate Pareto optimum, based on the different threshold designs. Numerical results show that the performance loss of the proposed algorithms is less than 10%, compared with the centralized optimal solution.**

I. Introduction

With the rapid development of modern wireless services and systems, wireless spectrum becomes increasingly scarce and valuable. However, recent measurements showed that most of licensed spectrum bands are heavily under-utilized [1]. Dynamic spectrum sharing is a promising approach to solve the spectrum under-utilization problem by allowing secondary (unlicensed) users to opportunistically access the idle licensed spectrum without degrading the performance of primary (licensed) users.

Since secondary users may not belong to the same service provider, achieving efficient spectrum sharing among secondary users in a distributed fashion is quite challenging. Non-cooperative game theory is widely used to model the competitive spectrum sharing among selfish secondary users (e.g., [2], [3], [4] and the references therein). Nie and Comaniciu in [2] modeled the channel allocation as a potential game. Attar *et al.* in [3] used the idea of bargaining to design an efficient and fair scheme for spectrum sharing. Niyato and Hossain in [4] proposed a price-based model for spectrum access mechanism. A common assumption of most existing work is that secondary users have complete information of channel characteristics and know the actions of neighboring users by explicit information exchange. However, information exchange among secondary users may lead to significant communication overhead, and may not be feasible due to privacy and security issues. Thus a fully decentralized algorithm without explicit information exchange is highly desirable in practice.

Another critical issue of distributed spectrum sharing is how to take the spatial reuse into account. Spatial reuse is a key feature of wireless communications. When different secondary users are geographically separated, they may access the same spectrum without causing intolerable mutual interferences. How to utilize the spatial reuse to design efficient distributed spectrum sharing mechanisms is becoming increasingly important. Recently, Tekin *et al.* in [5] extended the classic congestion game to graphs to formulate the spatial spectrum access problem. Chen and Huang in [6] considered a general class of spatial congestion game, where user payoffs reflect common wireless protocols (instead of simple functions of the number of competing users). In the distributed learning algorithm proposed in [6], secondary users need to frequently switch channels even after the system converges to a mixed strategy equilibrium, which may lead to significant system overhead.

To address these challenges, we propose two threshold-based decentralized spectrum access algorithms for the distributed spectrum sharing problem with spatial reuse and without any information exchange. The main contributions of this paper are as follows:

- Threshold-based decentralized algorithm design: we propose two novel threshold-based decentralized spectrum access algorithms, which take the spatial reuse into consideration and do not require information exchange among secondary users. The proposed algorithms can reach an equilibrium where secondary users do not need to switch channels.
- Convergence analysis: for the first proposed algorithm where each secondary user has a channel-specific throughput threshold, we show that it converges to an approximate Nash equilibrium. For the second proposed algorithm where each secondary user has multiple thresholds, we show that it achieves an approximate Pareto optimum.
- Superior performance: Numerical results show that the performance loss of the algorithms is less than 10%, compared with the centralized optimal solution.

The rest of the paper is organized as follows. We describe the system model in Section II. We introduce the decentralized algorithm based on conservative throughput thresholds in Section III, and then extend it to the layered threshold based algorithm in Section IV. We demonstrate the algorithm performance through numerical results in Section V, and finally conclude in Section VI. Due to the page limit, all proofs are contained in the online technical report [7].

II. System Model

A. Network Model

We consider a spectrum sharing network with a set of channels denoted by $\mathcal{M} = \{1, 2, \cdots, M\}$. The mean data rate of a channel $m \in \mathcal{M}$ is B_m. A set of secondary users $\mathcal{N} = \{1, 2, \cdots, N\}$ want to utilize these M channels. When a channel m is occupied by one or more primary users, we can set the mean data rate $B_m = 0$. For a limited period of

Bangyi Zhu and Jianwei Huang are with Department of Information Engineering, The Chinese University of Hong Kong; Email:zbyminor@gmail.com and jwhuang@ie.cuhk.edu.hk. Xu Chen is with School of ECEE, Arizona State University; Email: xchen179@asu.edu. This work is supported by the General Research Funds (Project Number CUHK 412710 and CUHK 412511) established under the University Grant Committee of the Hong Kong Special Administrative Region, China.

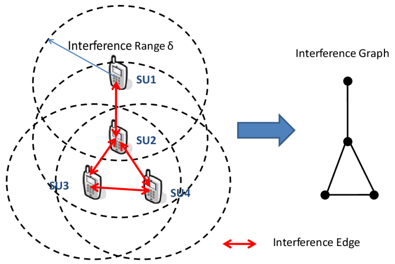

Fig. 1. Illustration of interference relationship among secondary users. SU1 (secondary user 1) and SU2 interfere with each other when SU1 and SU2 share the same channel.

time, the usage of spectrum by primary users is assumed to be static, *e.g.*, the activities of primary users of TV spectrum change very slowly. Similar as [6], we model the spatial reuse relationship among secondary users as a graph. We construct an undirected interference graph $G = (\mathcal{N}, \mathcal{E})$, where the vertex set $\mathcal{N}$ is the set of secondary users, and $\mathcal{E}$ is the set of interference edges among users (see Figure 1 for an example). A pair of secondary users n and n' are linked by an edge in the graph if and only if they are close enough to cause each other significant interference when they transmit concurrently on the same channel. We also denote user n's neighbors in the graph as $\mathcal{N}_n = \{i : (i, n) \in \mathcal{E}, i \in \mathcal{N}\}$.

B. Slotted Transmission Time Mechanism

We consider a time slotted transmission mechanism with the random backoff scheme [8]. In each time slot, each secondary user n executes the following three stages sequentially.

- Channel Contention: each secondary user randomly and uniformly chooses a number between 1 and λ_{max} and then counts down to zero. If there is no active transmission (by the other secondary users) on the chosen channel when the countdown timer expires, the secondary user will grab the channel. If multiple users contend for the same channel and have the same countdown number, there will be a collision and no one can grab the channel.
- Data transmission: each secondary user who successfully grabs its chosen channel transmits the data.
- Channel selection: each user chooses a channel to contend during the next time slot. This can be done based on the decentralized algorithms described in Sections III and IV.

Let $s_n(t)$ be the channel selected by secondary user n during the channel contention phase at time slot t, $\boldsymbol{s}(t) = (s_1(t), \cdots, s_N(t))$ be the channel selection profile for all secondary users, and $K_n^{s_n(t)}(\boldsymbol{s}(t))$ be the number of user n's neighbors who select the same channel $s_n(t)$. We can calculate the probability that secondary user n successfully grabs the channel $s_n(t)$ as:

$$
\begin{aligned}
& g(K_n^{s_n(t)}(\mathbf{s}(t))) \\
= & \lim_{\lambda_{\max}\to\infty} \sum\nolimits_{\lambda=1}^{\lambda_{\max}} \frac{1}{\lambda_{\max}} \left(\frac{\lambda_{\max}-\lambda}{\lambda_{\max}}\right)^{K_n^{s_n(t)}(\mathbf{s}(t))} \\
= & \frac{1}{K_n^{s_n(t)}(\mathbf{s}(t))+1}. \qquad (1)
\end{aligned}
$$

In the above calculation, we focus on the asymptotic case where λ_{max} goes to infinity, which can be a good approximation when λ_{max} is large. Based on (1), we can compute the expected throughput for each secondary user n as

$$u_n(\boldsymbol{s}(t)) = \frac{B_{s_n(t)}}{1 + K_n^{s_n(t)}(\boldsymbol{s}(t))}. \qquad (2)$$

Extending the proposed algorithms to the case of a finite λ_{max} is quite challenging, and will be considered in the future work.

C. Motivations for Threshold-based Algorithm Design

As mentioned before, in order to reduce the overhead in distributed spectrum sharing, it is desirable for secondary users not to exchange information and not to frequently change channels when arriving at a system equilibrium. Motivated by these practical considerations, we propose two novel threshold-based decentralized spectrum access algorithms. The key idea is to adaptively set the throughput threshold for each secondary user based on local observations. A user will only switch channel if his current throughput is lower than his threshold. Ideally, the throughput thresholds should satisfy three properties:

- *Local Computation*: A user can compute his threshold(s) with limited local information;
- *Feasibility:* The thresholds of all users can be reachable simultaneously;
- *Efficiency:* The total network throughput should be is high enough.

In the following analysis, we will use the terms " users" and "secondary users" interchangeably.

III. Decentralized Spatial Spectrum Access with Conservative Throughput Thresholds

A. Decentralized Algorithm Based on Conservative Throughput Thresholds

We first propose a threshold-based decentralized spectrum access algorithm based on users' conservative estimations of their environment. Intuitively, for a user n, the more users from his neighborhood compete with that user n on the same channel, the lower throughput that user n can possibly achieve. So we will define user n's throughput threshold based on the number of its neighbors, $d_n = |\mathcal{N}_n|$. For simplicity, we assume that each user n knows the number of neighbors d_n and the mean data rate B_m of each channel m[1]. As user n does not know the interference relationship among his neighbors, he will take the most conservative view and assume that all his neighbors are close-by and potentially interfere with each other when they select the same channel. In other words, user n assumes that his neighbors form a complete interference graph (see Figure 2 for an example). Under this conservative assumption, no two users can use the same channel without negatively affecting each other, and the maximum system

[1]The information of channel mean data rates and the number of neighbors can be measured based on the local observations such as the realized user throughput and channel contention history. Interested readers can refer to [6] and [8] for the detailed discussion.

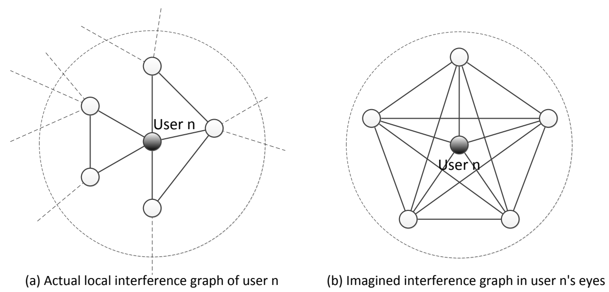

Fig. 2. The left subfigure (a) shows the actual interference graph in user n's neighborhood. The right subfigure (b) shows user n's imaginary local complete interference graph for the purpose of calculating the conservative throughput threshold.

throughput is $\sum_{j=1}^{M} B_m$ when all the channels are utilized. To achieve the maximum system throughput and a fair spectrum allocation at the same time, we would like to distribute the spectrum resources (measured in terms of throughput) equally among all the users on this complete interference graph, such that user n and each of his neighbors achieves the throughput $\sum_{m=1}^{M} B_m \Big/ (d_n+1)$. However, as each user can only just access one channel at a time and different channels have different throughputs, achieving such a fair allocation is in general very difficult. To resolve this issue, user n sets a channel-dependent threshold as follows to approximate a fair allocation.

Definition 1. ***(Conservative Throughput Thresholds (CTT))*** *Each user $n \in \mathcal{N}$ calculates a conservative throughput threshold E_n^m for each channel $m \in \mathcal{M}$ as follows:*

$$E_n^m = \frac{B_m}{\left\lceil (d_n+1)\, B_m \Big/ \sum_{j=1}^{M} B_j \right\rceil} \tag{3}$$

where $\lceil \cdot \rceil$ is the ceiling function.

Notice that the threshold is channel dependent. If we do not have the ceiling function in (3), then the channel independent threshold becomes $\sum_{m=j}^{M} B_m \Big/ (d_n+1)$, which may not be achievable as we have explained. The ceiling function enables each user to achieve the threshold on at least one of the channels. This throughput threshold is "conservative", because it is computed based on the assumption of a complete local interference graph. The actual throughput may be higher. Such a conservative threshold is a safe achievable choice without any information exchange.

Based on the conservative throughput thresholds (CTT), each user still needs to choose a proper channel to access by considering the coupling with other user choices. More precisely, a user's throughput on a channel is affected by the number of users choosing the same channel, but this number may change over time as every user tries to find a channel to satisfy his CTT. Thus users need to have a mechanism to learn the environment over time through measurements of the local achievable throughput and adjust the channel selection accordingly. In this regard, we employ the linear reward-penalty learning automata to update user n's channel selection $s_n(t)$ in a probabilistic manner as follows (here $p_n^j(t)$ denotes the probability of selecting channel j in each time slot t):

- If $u_n(t) \geq E_i^{s_n(t)}$

$$p_n^j(t+1) = \begin{cases} 1, & \text{if } j = s_n(t); \\ 0, & \text{otherwise.} \end{cases}$$

- If $u_n(t) < E_i^{s_n(t)}$

$$p_n^j(t+1) = \begin{cases} (1-b)p_n^j(t), & \text{if } j = s_n(t); \\ (1-b)p_n^j(t) + \frac{b}{M_n-1}, & \text{otherwise.} \end{cases} \tag{4}$$

Here the parameter $b \in (0,1]$. Intuitively, when user n chooses an unsatisfied channel that can not achieve his CTT on that channel, he will increase the probability of choosing other channels in the next time slot. The parameter b is related to the probability reduction for the unsatisfied channel.

We summarize the decentralized spatial spectrum access algorithm with CTT in Algorithm 1. Algorithm 1 will converge when no user wants to change his channel selection.

Algorithm 1 Decentralized Spatial Spectrum Access with Conservative Throughput Thresholds

```
initialization:
    each user n randomly chooses a channel s_n(t).
end initialization

loop for each time slot t and each user n ∈ N in parallel:
    if throughput u_n(t) < E_n^{s_n(t)} then
        randomly choose one channel from M_n channels based on distribution P_n(t) as in (4).
    else
        stick to the current channel selection.
    end if
end loop
```

B. Convergence Analysis

We now study the convergence of Algorithm 1. First, we show that there exists a channel selection profile such that all users can achieve their CTTs on their chosen channels simultaneously.

Theorem 1. *For any undirected interference graph G, there exists at least one channel selection profile $\boldsymbol{s} = (s_1, \cdots, s_n)$ such that for each user $n \in \mathcal{N}$, the achievable throughput u_n by selecting channel s_n is no smaller than the conservative throughput threshold $E_n^{s_n}$.*

Let $\tilde{\mathcal{S}}$ denote the set of channel selection profiles such that all users can achieve their CTTs on their chosen channels simultaneously. We can then prove the convergence of Algorithm 1.

Theorem 2. *From any initial channel selection profile $\boldsymbol{s}(0)$ and any parameter $b \in (0,1]$, we have*

$$\lim_{\tau \to \infty} Pr(\boldsymbol{s}(t+\tau) \in \tilde{\mathcal{S}} |\ \boldsymbol{s}(t) = \boldsymbol{s}(0)) = 1. \tag{5}$$

Theorem 2 indicates that Algorithm 1 will converge to a channel selection profile where all users' throughputs can satisfy their thresholds on the chosen channels simultaneously.

We next show that the proposed CTT-based decentralized spatial spectrum access algorithm can achieve an approximate Nash equilibrium. To proceed, we first define the general concept of ξ-approximate Nash equilibrium.

Definition 2. *A channel selection profile $\boldsymbol{s}^*$ is a ξ-approximate Nash equilibrium if*

$$u_n(s_n^*, s_{-n}^*) \geq \max_{s_n \in S_n} u_n(s_n, s_{-n}^*) - \xi, \; \forall n \in \mathcal{N}. \tag{6}$$

When $\xi = 0$, the approximate Nash equilibrium corresponds to the exact Nash equilibrium. We can show that the convergent channel selection profile of Algorithm 1 is an approximate Nash equilibrium.

Theorem 3. *The CTTs based decentralized spatial spectrum access algorithm in Algorithm 1 converges to a channel selection profile $\boldsymbol{s}^* = (s_1^*, \cdots, s_N^*)$, which is a ξ-approximate Nash equilibrium with*

$$\xi = \max_{n \in N} \left\{ \max_{s_n \in S_n} \left\{ B_{s_n} - E_n^{s_n^*} \right\} \right\}. \tag{7}$$

Numerical results in Section V show that Algorithm 1 can converge to a channel selection profile which is very close to a Nash equilibrium (with $\xi = 0$). Achieving a Nash equilibrium requires each user to have complete network information regarding all users' choices in all channels. Algorithm 1 does not require such information exchanging, and hence can only achieve an approximate Nash equilibrium.

IV. Decentralized Spatial Spectrum Access with Layered Thresholds

From the above analysis, we can see that CTT is often too conservative, and the overall network performance could be improved if we can adaptively select the CTT based on the actual achievable throughput history (especially when the actual interference grant is far from complete). This motivates us to introduce a threshold exploration mechanism into the algorithm design by considering CTT as a lower bound. In the new algorithm, each user has multiple thresholds and tries to select the best threshold based on long-term average throughput estimation. For the simplicity of algorithm design, the thresholds for each user are no longer channel-specific, but each user can set his thresholds based on his own demand and preference.

More specifically, each user n has a threshold set $D_n = \{d_1^n, d_2^n, \cdots, d_L^n\}$, where $d_1^n \geq d_2^n \geq \cdots \geq d_L^n$. Here d_L^n is the previously computed CTT value serving as a lower bound, i.e., $d_L^n = \min_{s_n \in S_n} \{E_n^{s_n}\}$, and d_1^n is the highest data rate of available channels, i.e., $d_1^n = \max_{m \in \mathcal{M}} B_m$. Without loss of generality, we assume that users all have L thresholds, but these thresholds can be different between users. A user n updates his the long-term average throughput as follows [9]:

$$\alpha_n(t+1) = \lambda \alpha_n(t) + (1-\lambda) u_n(t), \tag{8}$$

where $\lambda \in (0,1)$ is a smoothing parameter.

Each user n selects his threshold from the set D_n at every time slot t by considering the value of α_n. When $\alpha_n \in \left[d_l^n, d_{l-1}^n\right)$, user n will set d_l^n as the threshold (for all channels) and randomly select a channel to see whether the throughput satisfies the threshold. When the user cannot achieve the threshold, he will choose another lower threshold (i.e., the one immediately lower) to try again. If the smoothing factor λ is sufficiently close to 1 and the initial value $\alpha_n(0)$ is sufficiently high, then $\alpha_n(t)$ decreases very slowly and each user n hence needs to randomly explore different channel selections in order to set a proper threshold. Such a random channel exploration can lead to a better understanding the environment for each user and hence, as the time goes by, $\alpha_n(t)$ will indicate the achievable user throughput. This in turn will determine the proper threshold for each user. We summarize the layered thresholds based decentralized spatial spectrum access algorithm in Algorithm 2.

Algorithm 2 Decentralized Spatial Spectrum Access with Layered Thresholds

initialization:
  each user n sets the initial threshold as d_1^n and $\alpha_n \geq d_1^n$.
  each user n randomly chooses a channel $s_n(t)$.
end initialization

loop for each time slot t and each user $n \in \mathcal{N}$ in parallel:
  if $u_n(t)$ is lower than the current threshold **then**
    randomly choose one channel from M_n channels based on distribution $P_n(t)$ as in (4).
  else
    stick to the current channel selection.
  end if
  update α_n: $\alpha_n(t+1) = \lambda\alpha_n(t) + (1-\lambda)u_n(t)$
  if $\alpha_n \geq d_l^n$ and $\alpha_n < d_{l-1}^n$ for some $l \leq L$ **then**
    set d_l^n as the threshold.
  else if $\alpha_n < d_L^n$ **then**
    set d_L^n as the threshold.
  end if
end loop

Since we set the lowest CTT as the lower bound of the thresholds set for each user, the worst case performance of Algorithm 2 is the same as that of Algorithm 1. To study the performance of Algorithm 2, we introduce the concept of ε-Pareto optimum.

Definition 3. *The channel selection profile $\boldsymbol{s}'$ is ε-Pareto superior to the profile $\boldsymbol{s}$ if for any user n:*

$$u_n(s_n', s_{-n}') \geq u_n(s_n, s_{-n}) - \varepsilon, \forall n \in \mathcal{N}, \tag{9}$$

with the strict inequality for at least one user. The channel selection profile $\boldsymbol{s}$ is ε-Pareto optimal if there exists no other channel selection profile that is ε-Pareto superior to $\boldsymbol{s}$.

When λ is sufficiently close to 1, we can show Algorithm 2 converges to an ε-Pareto optimal solution.

Theorem 4. *Algorithm 2 has a sufficiently large probability p to converge to an ε-Pareto optimal solution with $\varepsilon = \max_{n \in \mathcal{N}} \{ \max_{1 \leq l \leq L} \{d_l^n - d_{l+1}^n\}\}$, if the parameter λ satisfies the*

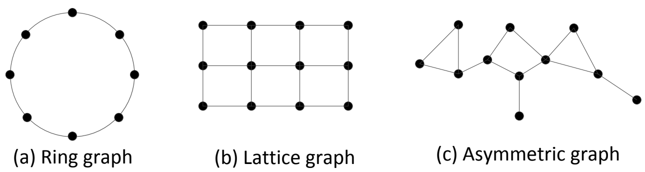

Fig. 3. Three types of interference graphs

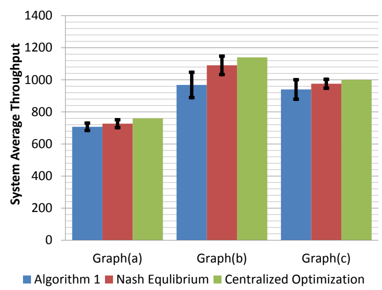

Fig. 4. Comparison of Algorithm 1, Nash equilibrium, and the centralized optimal solution

following condition:

$$\lambda \geq 1 - \frac{\max\limits_{n\in\mathcal{N}}\{\max\limits_{1\le l\le L}\{d_l^n - d_{l+1}^n\}\}\ln(1-\frac{1}{N\Delta^M})}{(B_1 - \frac{B_M}{\Delta})\ln(1-p)}, \quad (10)$$

where $\Delta = \max_{n\in\mathcal{N}}\{d_n\}$ *(i.e., the maximum degree of graph G).*

From Theorem 4, we can see that as the value of $\max_{n\in\mathcal{N}}\{\max_{1\le l\le L}\{d_l^n - d_{l+1}^n\}\}$ decreases, the performance Algorithm 2 can be further improved. However, more thresholds would push the λ closer to 1, which would increase the convergence time of Algorithm 2.

V. Numerical Results

In this part, we compare the performance of two proposed algorithms with the centralized global optimal channel selection profile and the Nash Equilibrium. We consider five channels in the network with mean data rates $\{100, 90, 70, 40, 15\}$ Mbps and three different interference graphs in Figure 3. We run the algorithms 100 times for each setting and then statistically summarize the results.

We first implement Algorithm 1 on three graphs in Figure 3. The simulations results are summarized in Figure 4, where we plot both the average value and the standard deviation error bars. The performance loss of Algorithm 1 is no more than 10% comparing to the centralized optimal solution. The performance gap between Algorithm 1 and Nash equilibrium is always within 5%.

We then evaluate the performance of Algorithm 2 , where each user n has L threshold that $D_n = \{d_1^n = 100, d_2^n, \cdots, d_L^n = \min_{s_n\in S_n}\{E_n^{s_n}\}\}$ with equal spacings. The simulation results with different values of L $(L = 1, 3, 5)$ together with the global optimal solution are shown in Figure 5. We can see the system average throughput approaches the centralized optimal solution as the number of thresholds increases. Figure 6 shows that Algorithm 2's convergence time also increases with the number of thresholds. This illustrates the tradeoff between performance and computation overhead.

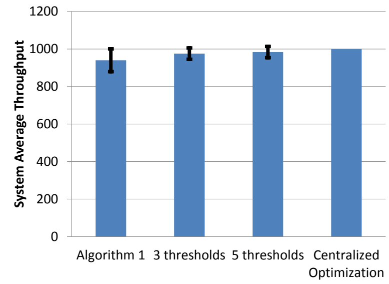

Fig. 5. Comparison of Algorithm 2 with different threshold settings and the centralized optimal solution in the graph (c)

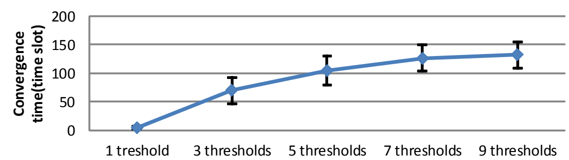

Fig. 6. The relationship between convergence time and number of thresholds on the graph (c) (in Fig. 3).

VI. Conclusion

In this paper, we design two threshold-based decentralized spatial spectrum access algorithms, which do not require explicit information passing among users and do not lead to frequent channel switchings after system convergence. As a result, the algorithms have low communication and computation overhead. Moreover, by properly choosing the number of layered thresholds, we can achieve a balance between the performance and the convergence time.

For future work, we are going to extend the results to other type of channel contention mechanism such as the slotted Aloha.

References

[1] FCC, "Second memorandum opinion and order," September 23, 2010. [Online]. Available: http://transition.fcc.gov/

[2] N. Nie and C. Comaniciu, "Adaptive channel allocation spectrum etiquette for cognitive radio networks," *Mobile Networks and Applications*, vol. 11, no. 6, pp. 779–797, 2006.

[3] A. Attar, M. R. Nakhai, and A. H. Aghvami, "Cognitive radio game for secondary spectrum access problem," *IEEE Transactions on Wireless Communications*, vol. 8, no. 4, pp. 2121–2131, 2009.

[4] D. Niyato and E. Hossain, "Competitive spectrum sharing in cognitive radio networks: a dynamic game approach," , *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, pp. 2651–2660, 2008.

[5] C. Tekin, M. Liu, R. Southwell, J. Huang, and S. H. A. Ahmad, "Atomic congestion games on graphs and their applications in networking," *IEEE Transactions on Networking*, 2012.

[6] X. Chen and J. Huang, "Spatial spectrum access game: Nash equilibria and distributed learning," in *ACM Mobihoc*, 2012.

[7] B. Zhu, X. Chen, and J. Huang, "Decentralized spatial spectrum access," Tech. Rep., 2013. [Online]. Available: http://jianwei.ie.cuhk.edu.hk/pulibcation/TRThreshold.pdf

[8] X. Chen and J. Huang, "Imitative spectrum access," in *IEEE Wiopt*, 2012.

[9] J. W. Crandall and M. A. Goodrich, "Learning epsilon-pareto efficient solutions with minimal knowledge requirements using satisficing," in *IEEE AAAI*, 2004.