Demand-Matching Spectrum Sharing in Cognitive Radio Networks: A Classified Game

Shaohang Cui and Jun Cai

Department of Electrical and Computer Engineering
University of Manitoba, Winnipeg, Manitoba, Canada
{shaohang,jcai}@ee.umanitoba.ca

Abstract. Cognitive Radio (CR) has been proposed as a promising technique to solve spectrum scarcity problem in wireless communications. For the implementation of CR, one major challenge is to design distributed spectrum sharing, which needs to efficiently coordinate CRs in accessing the spectrum opportunistically based on only local information. To address this problem, in this paper, we make use of the heterogeneity among users in cognitive radio networks (CRNs) and propose a distributed cooperative game with classified players. A prioritized CSMA/CA technique is adopted so that CRs select channels and their priority to access channel based on their satisfaction history, a public signal for CRs to collaborate to achieve the Correlated Equilibrium (C.E.). A no-regret learning algorithm is adopted to learn the C.E. Simulation results show that the proposed C.E. based classified game (CECG) can achieve up to 40% better performance compared to the unclassified one.

Keywords: Cognitive radio, distributed spectrum sharing, classified game, correlated equilibrium, no-regret learning.

1 Introduction

With more and more wireless services emerging in the market, the spectrum scarcity problem arises as the bottleneck for the future development of wireless communications. However, based on Mitola's research [1], most fixed allocated spectrum is severely under-utilized. Cognitive Radio (CR) which smartly utilizes spectrum is thus proposed as a promising technology to alleviate the increasing stress on the fixed and limited radio spectrum. In such networks, CRs are secondary users to the spectrum. Namely, they must obey certain interference constraints so that its transmission will not interfere the communication of Primary Users (PUs), the licensed users of the spectrum. In this way, they are envisioned to be aware of the physical environment and capable to adjust their transmission accordingly.

In cognitive radio networks (CRNs), a major challenges is to achieve the coordination among CRs to share the spectrum effectively. However, centralized approaches are deemed to be impractical in CRNs, due to the complexity and cost to setup a Common Control Channel and to exchange control information.

P. Ren et al. (Eds.): WICON 2011, LNICST 98, pp. 534–546, 2012.

Distributed approaches are thus proposed. The key issue in designing distributed spectrum sharing is that the decisions of spectrum allocation should be made independently by each radio based only on its own information. Some research works have been done in literature. In [2], a biologically-inspired algorithm was proposed, which enabled the CR to eventually learn the appropriate spectrum band and adapt the probability to select a channel. In [3], a non-cooperative game model was used to obtain the spectrum allocation among a primary user and multiple secondary users. The problem was formulated as an oligopoly market competition, and Nash equilibrium (N.E.) is considered as the solution of the game. The Correlated Equilibrium (C.E.), which is more general than the Nash equilibrium, was considered for dynamic spectrum access in [4] and [5]. In [4] and [5], CSMA/CA was adopted as the sharing technique, which allocated channel to CRs equally. However, by considering the heterogeneity of CRNs, in terms of the channel conditions, the application-based channel requirements among CRs, and the time-varying channel availability, sharing channel equally may result in low resource utilization efficiency.

Inspired by prioritized CSMA in IEEE802.11e [6] [7], in this paper, we introduce a priority to classify CRs to improve the network performance, in terms of the number of satisfied CRs by allocating different portion of the channel to CRs based on their demands. A new algorithm to estimate the number of CRs in different priority levels is also proposed. In the channel allocation process, each CR jointly determine its channel selection and priority based on its possible satisfaction and the loss it may introduce to other CRs. Such tradeoff between satisfaction and cost results in a distributed cooperative game which can maximize the satisfaction of the whole network. No-regret learning algorithm is adopted to reach the C.E. of the proposed game. Simulation results show that the C.E. based classified game (CECG) can achieve up to 40% better performance compared to the unclassified game in highly heterogeneous networks.

The rest of this paper is organized as follows: In Section 2, we present the system model and utility function. In Section 3, we study the C.E. and an no-regret learning algorithm. Simulation results are shown in Section 4 and finally conclusions are drawn in Section 5.

2 System Model

Consider an overlay CRN. The primary users have a strict priority on the spectrum access while CRs can only access spectrum not being utilized by PUs. As we focus on the competition and collaboration among CRs in spectrum sharing, we ignore the cost and faults from spectrum sensing. Namely, each CR is equipped with a perfect spectrum sensing technique, which can alway detect the presence of PUs instantly. We consider a simple CR transceiver which can be tuned in a wide range of spectrum, but can operate only on one channel at any time. All CRs are in the interference range of each other, and thus have to compete for the idle channels. CSMA/CA is used as the sharing technique. To improve the efficiency by considering network heterogeneity, we introduce priority mechanism to differentiate users with respect to their specific transmission requirements and channel qualities. Since applying multiple (> 2) priorities may introduce high complexity with marginal improvement on performance, as shown in simulation results, we consider two priority levels in our algorithm.

2.1 Network Structure

Assume that there are N channels in the system, represented as a channel set $\{C_N\}$. Each channel is licensed to a PU and total I CRs seek for channel access opportunistically. CRs belongs to two different classes, i.e., class 1 and class 2, with low and high priority to access channels, respectively. Time is divided into slots and we label them as $t = 1, 2, ...$ In a slot, both PUs' activities and CRs' strategies keep unchanged. Each CR's action consists of tow parts: channel selection and priority selection. At the beginning of any slot t, each CR i, $i = 1, 2, ..., I$, knows the following:

1) $r_i^{req} \in R^+$: the demand of CR i (in bits per time slot) to satisfy its QoS requirements, where R^+ denotes the set of positive real numbers.

2) $C_{i,n}^t \in R^+$: the channel quality in terms of transmission rate in bits per time slot for CR i on channel n at time t.

3) $A_{i,n}^t \in \{0, 1\}$: the availability of channel n for CR i at time t, which is determined by PUs' activities and the locations of both PUs and CR i. $A_{i,n}^t = 1$, if channel n is available for CR i at time t; otherwise, $A_{i,n}^t = 0$. $A_i^t = (A_{i,1}^t, ..., A_{i,N}^t)^T$ is the channel availability vector for CR i.

4) Ac_i^{t-1}: the action of CR i in the last slot $t-1$. $Ac_i^{t-1} = (X_i^{t-1}, P_i^{t-1})$ is chosen from the action set

$$\Omega_i^{t-1} = S_i^{t-1} \times Sp_i^{t-1} \tag{1}$$

In (1), S_i^{t-1} is the channel allocation decision space and can be represented as

$$S_i^{t-1} = \{X_i^{t-1} \in (0, 1)^c : {X_i^{t-1}}^T (1 - A_i^{t-1}) = 0, \sum_{n \in C_N} X_{i,n}^{t-1} \leq 1\} \tag{2}$$

where $X_i^{t-1} = (X_{i,1}^{t-1}, ..., X_{i,N}^{t-1})^T$ is the channel allocation decision of CR i. As indicated in (2), CR i can only select one available channel n with $X_{i,n}^{t-1} = 1$. Sp_i^{t-1} is the priority space of CR i

$$Sp_i^{t-1} = \{1, 2\} \tag{3}$$

We have $P_i^{t-1} \in Sp_i^{t-1}$. $P_i^{t-1} = 1$ stands for low priority, while $P_i^{t-1} = 2$ stands for high priority.

5) $r_i^{t-1} \in R^+$: the achieved average channel rate for CR i in the last slot $t-1$, which is determined by the number of CRs in the allocated channel and their priorities in the last slot, i.e., by CR i's action Ac_i^{t-1} and all other users' actions, denoted as Ac_{-i}^{t-1}. This data can be acquired from the amount of data transmitted in the last time slot.

6) $N1_n^{t-1*}$ and $N2_n^{t-1*}$: the estimated number of users of class 1 and class 2 in the last slot $t-1$ on the selected channel n, respectively. An estimation method will be discussed later.

Based on the aforementioned information, each CR i makes its decision Ac_i^t for slot t. Note that CRs make their decisions based on local information only, which allows decentralized algorithms.

2.2 Prioritized CSMA/CA

CRs share channels using a prioritized CSMA/CA scheme. By allocating less waiting time on average to CRs with higher priority, these CRs have a higher chance to capture the channel than others.

We introduce following definitions for protocol description:

1) subslot: the time needed for a CSMA attempt. We assume K subslots constitute a slot which are denoted as $t_1, t_2, ..., t_K$. Note that the length of subslots are not equal and so is the length of slots.

2) minislot: the time needed by CR to determine whether another station has accessed the medium.

3) SIFS (Short Interframe Space): the smallest period between packets. It has a duration at least enough for CR to sense the channel clear and switch between receiving and transmitting modes.

4) AIFS (Arbitration Interframe Space): the smallest waiting time before sending a packet. It depends on the corresponding priority class and is larger than SIFS.

5) RTS/CTS: Request to Send frame/ Clear to Send frame.

6) DATA/ACK: Data frame/ Acknowledgment frame.

7) CW: Contention Window which depends on the corresponding priority class.

Fig. 1 shows the protocol of prioritized CSMA/CA. As illustrated in the figure, in any subslot t_k, for CR i wishing to send data, it generates its backoff time $\tau_i(t_k)$ according to a uniform distribution within the interval $(0, CW[P_i^t])$. The backoff counter starts decreasing after detecting that the channel is idle for an $AIFS[P_i^t]$. Upon expiry of the backoff counter, the CR sending an RTS to initiate its data transmission if the channel is still sensed clear. Only one radio with the smallest waiting time $WT_i = \tau_i(t_k) + AIFS[P_i^t]$ will transmit successfully on

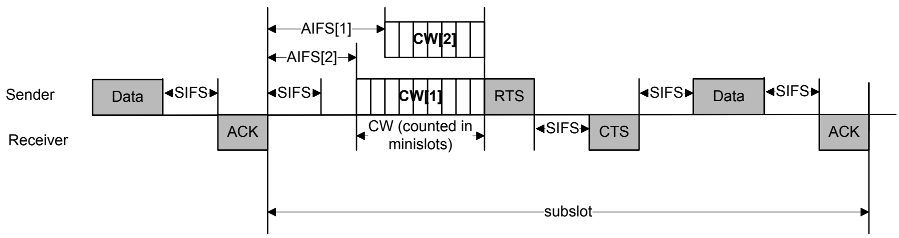

Fig. 1. Multiple backoff in Prioritized CSMA/CA

channel n in subslot t_k. The values of $AIFS$ and CW are set to guarantee that CRs in class 2 can have a smaller expectation of WT than those in class 1, so that they are more likely to take a smaller waiting time. Thus, CRs in class 2 have a higher priority to access the channel. The probability for CR i to catch channel n is

if $P_i^t = 2$

$$Prob_{i,n} = \tfrac{\delta}{CW[2]} \times \int_0^{\delta} (\tfrac{\tau}{CW[2]})^{N2_n - 1} d\tau + \int_{\delta}^{CW[2]} (\tfrac{\tau}{CW[2]})^{N2_n - 1} (\tfrac{\tau - \delta}{CW[1]})^{N1_n} d\tau \tag{4}$$

if $P_i^t = 1$

$$Prob_{i,n} = (1 - \frac{\delta}{CW[2]})^{N2_n} \times \int_0^{CW[1]} (\frac{\tau}{CW[1]})^{N1_n + N2_n - 1} d\tau \tag{5}$$

where $\delta = AIFS[1] - AIFS[2]$, and

$$N1_n^t = \sum_{i=1}^{I} \{X_{i,n}^t = 1\}\{P_i^t = 1\} \tag{6}$$

$$N2_n^t = \sum_{i=1}^{I} \{X_{i,n}^t = 1\}\{P_i^t = 2\} \tag{7}$$

Each CR will determine its own priority based on its utility, a function of its demand and satisfaction. The utility function will be discussed laster.

2.3 Decision-Feedback-Reaction Model

At the beginning of the t-th slot, each CR makes its decision based on the information about the network and its satisfaction, and hold this decision for the whole period of this slot. Note that channel catching probability and contention probability for CR i at slot $t-1$ are determined by all CRs on channel n, and they are known to CR i before slot t from channel catching results in the last slot. Hence, such probability can be seen as the feedback of CR i's action in the $(t-1)$-th slot. In realistic application, the number of subslots in a slot should be large enough to provide an accurate feedback. Based on this feedback, CR can make estimation of $N1_n^{t-1^*} and N2_n^{t-1^*}$, predict its future utility, and update its action in the next slot.

We introduce a simple estimation method for $N1_n^{t-1^*}$ and $N2_n^{t-1^*}$as follows.

For CR i, if $P_i^t = 2$, the probability for CR i to successfully catch the channel after waiting a period in the range of $(AIFS[1], AIFS[2])$ is

$$Pcat21_{i,n} = \frac{\delta}{CW[2]} \times \int_0^{\delta} (\frac{\tau}{CW[2]})^{N2_n - 1} d\tau \tag{8}$$

Obviously, $Pcat21_{i,n}$ is only determined by $N2_n$, and could be acquired from CR i's competition results. Hence, $N2_n^*$ can be estimated from $Pcat21_{i,n}$ by, for example, maximum-likelihood estimation [8]. Then, substituting $N2_n$ in (4), we can have the estimated value of $N1_n$.

Similarly, if $P_i^t = 1$, the probability for CR i to contend on the channel after waiting a period in the range of $(AIFS[1], AIFS[2])$ is

$$Pcon11_{i,n} = 1 - (1 - \frac{\delta}{CW[2]})^{N2_n} \quad (9)$$

which is also only determined by $N2_n$. Thus we can similarly estimate $N2_n^*$ from $Pcon11_{i,n}$ and then estimate $N1_n$ by substituting $N2_n$ in equation (5).

In this paper, accurate estimates of $N1_n$ and $N2_n$ are considered, i.e., $N1_n^* = N1_n$ and $N2_n^* = N2_n$. However, as shown in the simulation, even up to 30% estimation error will not affect the performance of the proposed algorithm significantly.

3 Optimization Problem and Game Formulation

For a scenario with strict QoS requirement, for instance, voice transmission, a meaningful global system object should aim to guarantee as many CRs' satisfaction as possible. Here, the satisfaction means that the achieved average rate should be no less than the required one. Hence, we adopted a utility function different from the best effort utility functions in [5] to better match the scenarios with strict QoS requirements. As a decentralized scheme is required, a local utility function is defined to guide the allocation decision of each CR. In follows, we will first introduce the global optimization problem, and then discuss the distributed game and utility function in details.

3.1 Global Optimization Problem

The global object is to maximize the number of satisfied users. As the optimization problem is held for any time t, we ignore the index t for simplicity. Let $Ac = (Ac_1, ..., Ac_I)$ be the joint action of all radios. The optimization problem can be formulated as:

$$\max_{Ac} \sum_{i=1}^{I} (G(r_i, r_i^{req})) \quad (10)$$

s.t.

$$Ac \in \Omega = \Omega_1 \times ... \times \Omega_I \text{ (joint action set of all radios)} \quad (11)$$

where

$$r_i = \sum_{n \in C_N} X_{i,n} r_{i,n} \leq 1 \quad (12)$$

$$r_{i,n} = Prob_{i,n} A_{i,n} C_{i,n} \quad (13)$$

$r_{i,n}$ is the achievable rate for CR i on channel n. $Prob_{i,n}$ is the probability for CR i to catch channel n, as defined in (4) and (5). $G(a,b)$ is a logic function to check whether CR i is satisfied, i.e.,

$$G(a,b) = \begin{cases} 1 & , \quad a \geq b \\ 0 & , \quad a < b \end{cases} \tag{14}$$

Note that once CR's QoS is satisfied, it has no intention to further increase its achievable rate.

3.2 Distributed Game and Local Utility

Each CR tries to access channel to satisfy its QoS requirements, while at the same time such access may cause loss to other CRs on the same channel, as it decreases other users' probability to catch the channel. Intuitively, if each CR tries to satisfy itself, and at the same time limits the loss it causes to other CRs, more CRs in the system could be satisfied. That is to say, CRs should select channels with good channel condition and less users on it. Thus for a cooperative distributed game which aims to improve the global performance, the local utility for each CR should be a tradeoff between its satisfaction and other CRs' loss. Then, from the game theory point of views, the satisfaction acts as the income while other CRs' loss as the price.

Note that the local utility function is only an estimation from the last slot. For instance, since the reward of each CR's action is determined by other CRs' actions, the estimated achievable average rate calculated at the beginning of a slot may differ from the exactly achieved one. However, our simulation indicates that the proposed algorithm converges after a number of rounds.

We define a distributed game as follows:

CRs are players in the game. Ac_i^t, the action of CR i in slot t, is selected from action set Ω_i^t defined in (1). Since any CR's utility is determined not only by itself but by other CRs' actions, the local utility for CR i is defined as:

$$U_i(Ac_i^t, Ac_{-i}^t) = U_i^1(Ac_i^t) + \alpha U_i^2(Ac_i^t) \tag{15}$$

where Ac_{-i}^t represents all other CRs' action.

In (15)

$$U_i^1(Ac_i^t, Ac_{-i}^t) = G(r_i^t, r_i^{req}) \tag{16}$$

stands for the satisfaction, where r_i^t is defined in (12), and

$$U_i^2(Ac_i^t, Ac_{-i}^t) = -P_i^t(\alpha_1 N1_n^{t-1^*} + \alpha_2 N2_n^{t-1^*}) \tag{17}$$

stands for the cost, i.e., the loss of all other users in the channel n with $X_{i,n} = 1$. Since it is hard to learn the real decrement on the channel rates for other users, a rough estimation is adopted. Note that if CR i choses to act with higher priority, it may induce more loss to all other CRs in the same channel, and thus it should pay more. Thus, if a CR can be satisfied with low priority, there is no motivation for it to select the high priority in the same channel.

In (15) and (17), $\alpha, \alpha_1, \alpha_2$ are user-defined tradeoff factors. Since the actual effect of CR i's action on the global utility is unknown, these weights are adjustable.

4 Correlated Equilibrium and No-Regret Learning

In this section, we adopt the concept of C.E. and introduce a no-regret learning algorithm as a distributed adaptive learning algorithm to solve the optimization problem defined in the previous section.

4.1 Correlated Equilibrium (C.E.)

A C.E. is a solution concept that is more general than the well known N.E. [9]. Given a public signal (in this paper, that is the satisfaction history of CRs), a strategy consists of recommendatory actions to every possible observation of the public signal a player can make. Thus, strategies of users are related to the public signal. Players reach the C.E. if no player would want to deviate from a recommended strategy. Note that N.E. corresponds to the special case of a C.E. The C.E. considers the interaction among players to make decision and thus could achieve better performance than N.E..

In the proposed distributed game, the C.E. is defined as: if and only if, for all player i, with $Ac_i \in \Omega_i$ as its action, a probability distribution $Pr(Ac_i, Ac_{-i})$ satisfies

$$\sum_{Ac_{-i} \in \Omega_{-i}} Pr(Ac_i, Ac_{-i})[U_i(Ac_i', Ac_{-i}) - U_i(Ac_i, Ac_{-i})] \leq 0, \quad \forall Ac_i', Ac_i \in \Omega_i \tag{18}$$

where $Pr(Ac_i, Ac_{-i})$ is the correlated strategy.

4.2 No-Regret Learning

No-regret learning (also called regret tracking, regret matching) is a kind of adaptive learning algorithms with fast convergence [10]. In no-regret learning, the probability to conduct an action is proportional to the regret for not having played other actions, and the stationary solution of the learning algorithm exhibits no regret. This algorithm will almost surely converge to C.E., as proved in [10].

For the action of CR i in slot t, $Ac_i^t \in \Omega_i^t$, we denote actions in the state space as $j \in \{0, 1, 2, ..., 2N\}$ for simplicity, i.e.: if $\exists X_{i,n}^t = 1$, $j = 2n + P_i^t - 2$; otherwise $j = 0$.

Each CR i executes the following steps:

1) Initialize arbitrarily probability of taking action for CR i. Set $\theta^{i,0} = 0$.

2) Generate regret matrix H^i with elements

$$H^i_{jk} = I\{Ac^t_i = j\} \times (U_{i,n}(k, Ac^t_{-i}) - U_{i,n}(j, , Ac^t_{-i})) \tag{19}$$

which stands for the regret of not using action k, other than the real action j, in slot t.

3) Set a regret value

$$\theta^{i,t+1}_{jk} = \theta^{i,t}_{jk} + \epsilon(H^i_{jk} - \theta^{i,t}_{jk}), 0 < \epsilon << 1 \tag{20}$$

which stands for the average gain that CR i would have received had he chosen action k in the past (from time 0 to t) instead of j. Here, ϵ is the learning rate.

4) Update action

CR i updates action $Ac^{t+1}_i = k$ with probability

$$P(Ac^{t+1}_i = k|Ac^t_i = j) = \begin{cases} \max(\theta^{i,t+1}_{jk}, 0)/\mu_i & , \quad k \neq j \\ 1 - \sum\limits_{i \neq j} \max(\theta^{i,t+1}_{jk}, 0)/\mu_i & , \quad k = j \end{cases} \tag{21}$$

In (21), μ_i is an arbitrary updating rate that is sufficiently large, i.e.,

$$\mu_i > (N_{Ac_i} - 1)(u^{\max}_i - u^{\min}_i) \tag{22}$$

where N_{Ac_i} is the number of actions for CR i, $u^{\max}_i$ is the maximum achievable utility, and $u^{\min}_i$ is the minimum utility for CR i. In our work, we set $\mu_i = (N_{Ac_i} + 1)(u^{\max}_i - u^{\min}_i)$.

Note that the algorithm requires that CR i knows what utility it would have received for each action, even if that action was not taken. This puts a request to know the number of users of each class on each channel. In fact, a modified regret tracking algorithm can be used without such information [5] [11]. However, the convergence is far too slow.

5 Simulation Results

We focus on slightly congested systems, with total capacity of channels slightly less than the total user demand to highlight the effect of spectrum sharing algorithms on the resource utilization efficiency. For each CR, some randomly selected channels are set to be unavailable to reflect the occupation of PUs. For CRs' channel condition and required rate, we adopt randomly generated data following Gaussian distribution for simplicity to introduce heterogeneity among CRs.

In simulations, AIFS[1]=150, CW[1]=100, AIFS[2]=100, CW[2]=150, all in unit of minislots. In the case that there is only one user in each class on the same channel, the probability to catch channel for user in class 1 is 0.32, and for class 2 is 0.65. Learning rate $\epsilon = 0.1$, tradeoff factor $\alpha = 0.015$, $\alpha_1 = 1.1$ and $\alpha_2 = 2$ are obtained from simulation results.

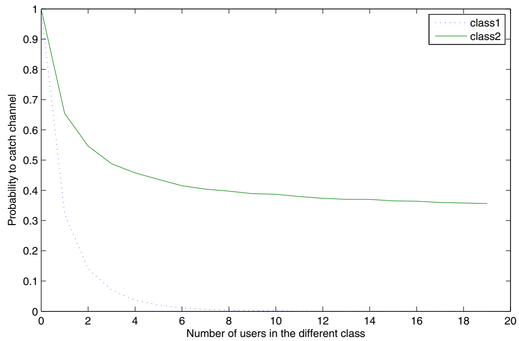

Fig. 2. Catching Probability vs. num of users of different class, with 1 user in the discussing class

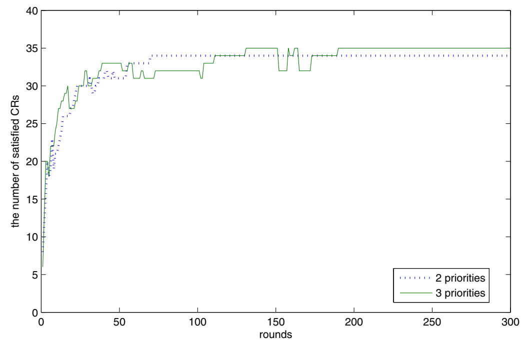

Fig. 3. $N = 3$, $I = 50$. Channel condition follow a Gaussian distribution with mean 30, variance 7. Required rate follow a Gaussian distribution with mean 3, variance 3.

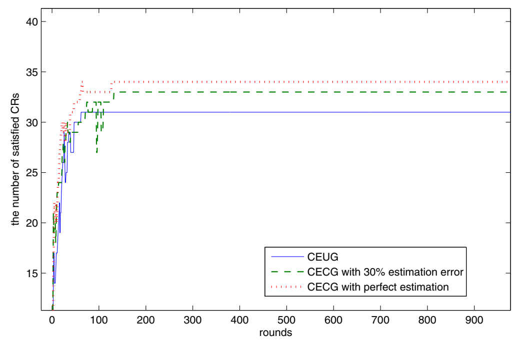

Fig. 4. $N = 3$, $I = 50$. Channel condition follow a Gaussian distribution with mean 30, variance 7. Required rate follow a Gaussian distribution with mean 3, variance 3.

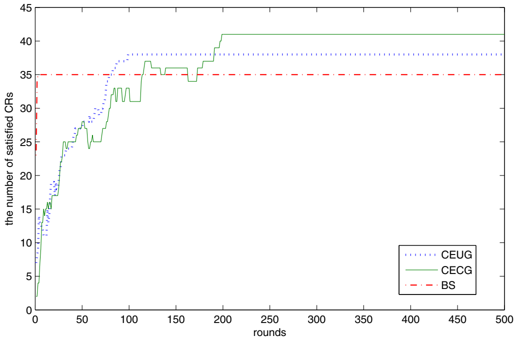

Fig. 5. $N = 15$, $I = 50$. Channel condition follow a Gaussian distribution with mean 15, variance 7. Required rate follow a Gaussian distribution with mean 7, variance 3.

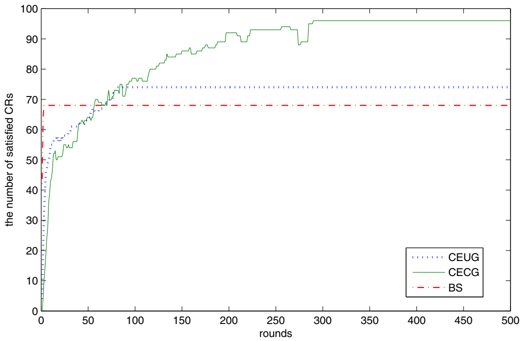

Fig. 6. $N = 25$, $I = 100$. Channel condition follow a Gaussian distribution with mean 15, variance 7. For 50 Users, the required rate follow a gaussian distribution with mean 7, variance 3, and other 50 Users with with mean 3, variance 3.

Fig. 2 compares the catching probability and sensitivity of users in different classes. From this figure, we can see that for a user in class 1, if we increase the number of users in class 2, the catching probability decreases rapidly; while for the user in class 2, if we increase the number of users in class 1, the probability decreases much slower and converges to a non-zero limitation. That is because, for the catching probability of class 2, the first part in equation (4) is not affected by $N1_n$, and the second part in (4) converges to 0 with large $N1_n$; while for catching probability of class 1, with $N2_n$ in the exponent, it decreases rapidly with increasing $N2_n$.

Fig. 3 compares the performance if 3 other than 2 priority levels are applied. For the 3-priority level case, we set AIFS[1]=150, CW[1]=100, AIFS[2]=125, CW[2]=125, AIFS[2]=100 and CW[2]=150. From this figure, we can see that with 3-level priority, the algorithm can only provide marginal performance improvement, but much slower convergence rate. That is because more priority levels will introduce a larger action set, which increases the complexity. Moreover, as in the proposed sharing algorithm, an unsatisfied CR can switch to the channel with better channel condition and lighter competition to increase its throughput other than continuously increasing its priority in the same channel, the improvement from more priorities becomes insignificant. This justifies our selection of 2 priority levels.

Fig. 4 shows the influence of estimation error. The performance of the C.E. based unclassified game (CEUG) in [5] is adopted as a comparison benchmark. From this figure we can see that for up to 30% estimation error, the performance of our algorithm is just affected slightly. The reason is that users are dispersed in all actions and the number of users with the same action is not large. Thus, the estimation error can only change the number of users with the same action slightly.

Fig. 5 compares the performance of the proposed CECG algorithm with CEUG in [5]. The best response (BS) algorithm with unclassified game in [12] is also adopted for comparison. In the BS algorithm, in every round each CR selects the channel with largest utility, and it has been proved in [12] that the N.E. of this unclassified game can be achieved. From the figure, performance improvement can be obviously observed in terms of the number of satisfied users. The introduction of C.E. brings in about 10% improvement comparing to the BS algorithm, as it considers the cooperation among CRs, at the cost of convergence rate. Note that if all CRs chose to be in the same class, our algorithm will degrade to that in [5]. Comparing to [5], since the proposed algorithm has a larger action set including those in CEUG, at least we can acquire a same performance as the CEUG algorithm.

Fig. 6 further demonstrates the influence of heterogeneity of users on the performance where two groups of CRs with difference in demands are applied. Comparing the results in Fig. 5 and Fig. 6, we can find that the improvement of CECG over CEUG (about 40%) is larger in the latter case than that in the former (about 10%). The simulation results further justify the necessity to apply the proposed algorithm for performance improvement in CRNs, especially when significant heterogeneity exists among CRs.

6 Conclusion

By taking into account the heterogeneity among users in CRNs, we proposed a distributed cooperative game with classified players in this paper for efficient spectrum sharing, where CRs select channel and their priority based on their satisfaction history. This satisfaction history is used as a public signal for CRs to collaborate with each other to achieve the C.E. A no-regret learning algorithm is adopted to learn the C.E. Simulation results show that the classified game has a better performance compared to the unclassified game, and the improvement is determined by the heterogeneity of the network.

Acknowledgment. This work is supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada IRC and Discovery Grants.

References

1. Mitola III, J.: Cognitive radio: an integrated agent architecture for software defined radio. Ph.D. Thesis, KTH Royal Inst. Technology, Stockholm, Sweden (2000)
2. Atakan, B., Akan, O.B.: BIOlogically-inspired spectrum sharing in cognitive radio networks. In: Wireless Communications and Networking Conference, WCNC 2007, pp. 43–48. IEEE (2007)
3. Liu, J., Shen, L., Song, T., Wang, X.: Demand matching spectrum sharing game for non-cooperative cognitive radio network. In: International Conference on Wireless Communications & Signal Processing, WCSP 2009, pp. 1–5 (2009)
4. Zhu, H., Pandana, C., Liu, K.J.R.: Distributive Opportunistic Spectrum Access for Cognitive Radio using Correlated Equilibrium and No-Regret Learning. In: Wireless Communications and Networking Conference, WCNC 2007, March 11-15, pp. 11–15. IEEE (2007)
5. Maskery, M., Krishnamurthy, V., Qing, Z.: Decentralized dynamic spectrum access for cognitive radios: cooperative design of a non-cooperative game. IEEE Transactions on Communications 57(2), 459–469 (2009)
6. Mangold, S., Choi, S., May, P., Klein, O., Hiertz, G., Stibor, L.: IEEE 802.11e Wireless LAN for Quality of Service. In: Proc. IEEE European Wireless 2002, Florence, Italy (February 2002)
7. Blake, S., et al.: An Architecture for Differentiated Services, RFC 2475 (December 1998)
8. Weisstein, E.W.: Maximum Likelihood. From MathWorld–A Wolfram Web Resource, `http://mathworld.wolfram.com/MaximumLikelihood.html`
9. Aumann, R.J.: Subjectivity and correlation in randomized strategy. Journal of Mathematical Economics 1(1), 67–96 (1974)
10. Hart, S., Mas-Colell, A.: A simple adaptive procedure leading to correlated equilibrium. Econometrica 68(5), 1127–1150 (2000)
11. Hart, S., Mas-Colell, A.: A reinforcement procedure leading to correlated equilibrium. In: Economic Essays, pp. 181–200. Springer, Heidelberg (2001)
12. Pillutla, L.S., Krishnamurthy, V.: Game Theoretic Rate Adaptation for Spectrum-Overlay Cognitive Radio Networks. In: Global Telecommunications Conference, IEEE GLOBECOM 2008, November 30-December 4, pp. 1–5. IEEE (2008)