All-Powerful Learning Algorithm for the Priority Access in Cognitive Network

M. Almasri
LABSTICC, UMR 6285 CNRS
ENSTA Bretagne
29806 Brest Cedex 9, France
mahmoud.almasri@ensta-bretagne.org

A. Mansour
LABSTICC, UMR 6285 CNRS
ENSTA Bretagne
29806 Brest Cedex 9, France
mansour@ieee.org

C. Moy
IETR - UMR 6164 CNRS
Univ Rennes
F-35000, Rennes, France
christophe.moy@univ-rennes1.fr

A. Assoum
Faculté des sciences
Université Libanaise
Tripoli, Lebanon
a.assoum@ul.edu.lb

C. Osswald
LABSTICC, UMR 6285 CNRS
ENSTA Bretagne
29806 Brest Cedex 9, France
christophe.osswald@ensta-bretagne.fr

D. Le Jeune
LABSTICC, UMR 6285 CNRS
ENSTA Bretagne
29806 Brest Cedex 9, France
denis.le_jeune@ensta-bretagne.fr

***Abstract*—In this paper, we propose the All-Powerful Learning (APL) algorithm for multiple Secondary Users (SUs) that considers the priority access and the dynamic multi-user access, where the number of SUs changes over time. To the best of our knowledge, APL is the first learning algorithm that successfully handles the dynamic users with the priority access. APL does not require any cooperation or prior information (e.g. the number of users in the network, or the number of available channels, or the total number of iterations) as do many existing algorithms. We should emphasize that the knowledge of previous parameters can make all these algorithms impractical and difficult to apply. The experimental results show the superiority of APL compared to existing algorithms.**

***Index Terms*—Multi-Armed Bandit, Priority Access, Competitive Network , Opportunistic Spectrum Access, All-Powerful Learning Algorithm, Cognitive Network.**

I. Introduction

The explosive growth of wireless services and applications during the past 30 years illustrates the increasing demand of communications and resources. To tackle the static spectrum allocation problems, the Cognitive Radio (CR), firstly proposed by Mitola [1], has been proposing several solutions for Dynamic Spectrum Access (DSA). One of them is Opportunistic Spectrum Access (OSA), where Secondary Users (SUs: unlicensed users) are allowed to search, identify and exploit the available spectrum let free by the licensee of the spectrum band, e.g. Primary Users (PUs: licensed users), while limiting interference with the PUs. In OSA, a SU tries to identify spectral white spaces vacated by PUs when are not active. For strategic and logistic reasons as well as to simplify the complexity of SU receivers in our working context, we assume that the SU is able to sense and explore one channel at each time slot to find transmission opportunities. OSA in cognitive radio networks (CRNs) presents new challenges comparing to current wireless networks:

- Detection the activities of PUs: Since, SU should perform a spectrum sensing operation before transmitting, that can be achieved for instance by an energy detection [2].
- Sensing a wide radio bandwidth: Due to hardware constraints, the processing time and energy costs of spectrum detection, it is impractical for SU to scan all the channels at each time slot. Therefore, under a partially observation (one channel/slot), SU must select a channel to sense in the time interval and decide whether the detected channel is free to transmit his data.
- Sharing the white space among SUs while avoiding radio interference with the PUs: Under the multi-user case, two learning models can be considered to manage the secondary network: Cooperative or competitive learning. In our previous work, we proposed a cooperative learning algorithm taking into account the priority access [3]. A cooperative network can provide necessary information to learn the channels' availability and reduce the collision among users. Although, this can increase the exchanged information and the complexity of the network. In competitive learning scenarios, SUs access selfishly the channels without any constraint or information exchange with each other, and they are not subject to any central control.

By focusing on the OSA problem, we propose hereinafter a competitive learning algorithm to manage the secondary network. Our algorithm achieves a logarithmic regret (i.e. the loss of reward of SUs due to non-selection of best channels). By minimizing the regret, we can increase the transmission opportunities for the cognitive users. In our simulations, we evaluate the performance of our algorithm by showing: the global regret and the percentage of time the best channels have been used.

II. Multi-Armed Bandit Learning Algorithms

Due to its generic nature, the MAB problem takes a fundamental importance in stochastic decision theory and its applications can be traced in many engineering problems, such as: wireless channel access, jamming communication and

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

object tracking. The MAB problem can be roughly expressed as follows: an agent in front of slot machines must decide which machine to play at each time. Each machine has an average reward unknown to the user. The user goal is to find the best machine with the highest average reward, to maximize his cumulative gain. Generally, a good strategy must make a trade-off between the exploitation (using the machine with the highest known reward) or exploration (testing another machine trying to win more). Therefore, several learning algorithms have been proposed to solve the MAB issue, such as: TS [4], UCB [5] and $\epsilon - greedy$ [6]. It should be noticed that this learning approach is particularly suitable to the OSA problem, where the SU does not have any prior knowledge on its environment. Supposing that the time is slotted and the K independent identically distributed (i.i.d.) channels are ordered according to their availabilities, i.e. $\mu_1 > \mu_2 > \ldots > \mu_K$, and that one SU is considered. This latter can sense one channel at each time slot and send his data if the channel is free. Let $T_i(t)$ be the number of times the $i-th$ channel is sensed by the user up to time t and $r_i(t)$ represents the reward obtained from the $i-th$ channel at instant t. Therefore, we define the regret (i.e. the loss of reward due to selecting sub-optimal channels) as follows:

$$R(n, \beta) = n\mu_1 - E\left(\sum_{t=1}^{n} \mu_i^{\beta}(t)\right) \tag{1}$$

where n is the total number of iterations and $\mu_i^{\beta}(t)$ represents the vacancy probability of the selected channel at instant t using the learning algorithm β. The regret can also illustrate the performance of any MAB learning algorithm. The well-known and widely used MAB algorithms are:

- Thompson Sampling: represents the earliest learning algorithm, where the agent selects at each time slot the channel that has the highest index θ_i:

  $$\theta_i(t) = \frac{S_i(t) + a_i}{S_i(t) + a_i + F_i(t) + b_i} \tag{2}$$

  a_i, b_i are constant numbers and $S_i(t)$, $F_i(t)$ represent respectively the success and failure counts. If the channel is free, then we increase $S_i(t)$: $S_i(t) = S_i(t) + 1$ otherwise $F_i(t)$ = $T_i(t)$ $-S_i(t)$.
- Upper Confidence Bound: the first version of this algorithm is proposed in [5], his index $B_i(t)$ contains two variables $X_i(t)$ and $A_i(t)$ representing the exploitation (or the expected of reward) and exploration factors respectively:

  $$B_i(t) = X_i(t) + A_i(t) \tag{3}$$

  where: $X_i(t) = \frac{1}{T_i(t)} \sum_{j=1}^{t} r_i(j)$ and $A_i(t) = \sqrt{\frac{2\ln(t)}{T_i(t)}}$
- $\epsilon - greedy$: in this algorithm, the user selects a random channel if χ (i.e. a uniform random variable $\in$ [0,1]) $<$ ϵ_t where ϵ_t = min $\{1, \frac{H}{t}\}$ and H is a constant number, else SU selects the channel with the highest expected of reward $X_i(t)$.

III. Multi-User Learning Algorithm

The learning algorithms presented and discussed in the previous section are proposed in the case of a single OSA user as first proposed in [7] with UCB. Under the multi-user case, if each user applies a learning algorithm to find the best channel, i.e. to select the channel that has the highest index, then a very large number of collision could be happen since all the users try to reach the best channel. Subsequently, a distributed learning algorithm is required to manage the network of multiple SUs and decrease the number of collision among them.

A. Related Work

1) Priority Access: Many recent studies have been proposed for multiple SUs to take into account the priority access such as SLK [8], and $kth-MAB$ [9]. However, to best of our knowledge, all these algorithms do not consider the dynamic access in which a dedicated channel of a leaving user can't be used by the other users, as shown in Fig. 1. By taken into account the priority access, the authors of [9] proposed a learning algorithm where each user has a prior knowledge about his rank. In this algorithm, the time is slotted and each slot divided into multi sub-slot depending on the user priority ranks, i.e. the slots of SU_U is divided into U sub-slots in order to find the $U - th$ best channel and transmit the data via this channel. Therefore, the transmission time under a large number of users tends towards zero for the high ranking users, which is a major limitation of this algorithm. Based on UCB, the authors of [8] proposed the SLK algorithm that is an efficient algorithm for the priority access. However, the number of users must be fixed and known for each user.

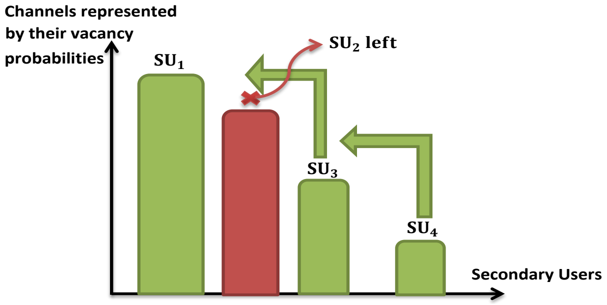

Fig. 1. Priority access after a user left his dedicated channel

2) Random Access: In the literature, several learning algorithms have been proposed for random access where the SU chooses randomly one of the best channels. In [10], the authors proposed the Musical chair algorithm and the Dynamic Musical Chair (for the dynamic access where users can enter into and out of the network). In the latter algorithms, each user selects a random channel up to a fixed time T_0 in order to estimate the channel availabilities and the number of users, U, in the network. After the time T_0, each user should select a random channel between $\{1, ..., U\}$.

To find the U-best channels, the authors of [11] proposed the Multi-user $\epsilon - greedy$ collision Avoiding (MEGA) algorithm based previously on the $\epsilon - greedy$ algorithm proposed in [6]. However, their algorithm suffers the same drawbacks of the Musical chair and Dynamic Musical Chair and does not consider the priority access.

B. All-Powerful Learning Algorithm (APL)

In our work, we are interested in the priority access where the SUs should access the channels based on their priority ranks. Our goal is to ensure that the U users are accessing separately the U-best channels. Our blind approach does not require any prior information to identify best channels. To the best of our knowledge, existing learning algorithms [8]–[12] for the multi-user access may suffer at least one of the following drawbacks:

1) The number of users should be a known constant by all users.
2) SUs should have a prior knowledge about the number of available channels.[1]
3) Total number of iterations or transmission time should be known for users.
4) The dynamic access is not allowed; under a dynamic access, any SU can at any instant join or leave the network.
5) A restricted dynamic access is considered, where a SU can't leave the network during the learning or the exploration phases.
6) The estimation of the dynamic channel availability is not a option; Therefore, the vacancy probability should be fixed.
7) An access priority among SU is seldom considered in the literature. Generally SU can select any channel.

For all these reasons, we propose in this section All-Powerful Learning algorithm (APL) in order to tackle the above mentioned drawbacks. Furthermore, we are interested in the dynamic priority access where the ranked users can join or leave the network at any time. In [8] and [9], the authors proposed SLK (Selective learning of the K-th largest expected rewards) and $kth - MAB$ learning algorithms for the priority access without considering the dynamic access. The latter algorithms also suffer some of the above drawbacks (mainly the 1^{st}, 2^{nd} and 4^{th} drawbacks).

In the classical method of priority access, the first priority user SU_1 should sense and access the best channel, μ_1, at each time slot. While the target of the second priority user SU_2 is to access the second best channel. To reach his goal, SU_2 should sense to find the two best channels at the same time, i.e. μ_1 and μ_2, in order to compute their availabilities and then access the second best channel if available. For the $U-th$ SU, he should estimate the vacancy probability of all the U first best channels at each time slot to access the $U-th$ best one. However, it is a costly and impractical method to settle down each user to his dedicated channel. In the case of APL, at each time slot the user can sense one channel and transmit his data if available (see algorithm 1). In our algorithm each SU_k has a fixed rank, $k \in \{1, ..., U\}$, and his target remains the access of the $k-th$ best channel. The major problem of the competitive priority access is that each user needs to selfishly estimate the channels vacancy probability as soon as possible in order to access his dedicated channel. Our algorithm can solve this problem by making each user generates a rank around his prior rank to have information about the channel availabilities. In this case, SU_k can scan the k best channels and his target is the $k-th$ best one. However, if the generated rank of SU_k is different to k then he accesses a channel of the set $\{\mu_1, \mu_2, ..., \mu_{k-1}\}$ and he may collide with top priority users, i.e. $SU_1, SU_2, ..., SU_{k-1}$. After each collision, the user can regenerate his restricted rank to access his assign channel; Otherwise, he retains his rank. However, if SU_k regenerates his rank at every slot, there will be a large collision number and all the transmission will be lost. Thus, after a finite number of iterations, each user settles down to his dedicated channel.

Algorithm 1: All-Powerful Learning algorithm

```
Input: $k, \xi_k(t)$, $r_i(t)$,
$k$: indicates the $k-th$ user or $k-th$ best channel,
$\xi_k(t)$: indicates a presence of collision for the $k-th$
  user at instant $t$,
$r_i(t)$: indicates the state of the $i-th$ channel at instant
  $t$, $r_i(t)$ = 1 if the channel is free and 0 otherwise,
Initialization
$k$ = 1,
for $t$ = 1 to $K$ do
    $SU_k$ senses each channel once,
    $SU_k$ updates his index $\theta_i(t)$, $B_i(t)$ or $X_i(t)$,
    $SU_k$ generates a rank of the set $\{1, ..., k\}$,
    $k+1$,
for $t$ = $K$+1 to $n$ do
    $SU_k$ senses a channel in his index $\theta_i(t)$, $B_i(t)$ or
      $X_i(t)$, according to his rank,
    if $r_i(t)$=1 then
            $SU_k$ transmits his data,
        if $\xi_k(t)$=1 then
            $SU_k$ regenerates his rank of the set
              $\{1, ..., k\}$,
        else
            $SU_k$ keeps his previous rank,
    else
        $SU_k$ refrains from transmitting at instant $t$,
    $SU_k$ updates his index $\theta_i(t)$, $B_i(t)$ or $X_i(t)$.
```

[1] In fact, this information is required when we use our algorithm under UCB or $\epsilon - greedy$ but it is not necessary in the case of Thompson Sampling. However, in the case of UCB the user should access each channel once in the initialization part in order to have a prior information about the channel availabilities. As well as, in the case of $\epsilon - greedy$ the constant H introduced in section II depends on the number of channels K.

IV. Simulations and Results

In this section, we evaluate the performance of our algorithm for an unknown static number of users $U \leq K$ as well as in the scenario of dynamic operation where the number of users is unknown and can change as users may join or abandon the network. We begin with the static setting where the number of users equals 4 (U=4). We assume that the network contains 9 orthogonal channels (K=9) with the following mean reward:

$$\mu = [0.9 \;\; 0.8 \;\; 0.7 \;\; 0.6 \;\; 0.5 \;\; 0.4 \;\; 0.3 \;\; 0.2 \;\; 0.1]$$

The μ vector is initially unknown to the user. After estimating the availabilities of the communication channels, the targets of the users SU_1, SU_2, SU_3 and SU_4 are the respective access to the 4 best channels, (i.e. $\mu_1 = 0.9$, $\mu_2 = 0.8$, $\mu_3 = 0.7$ and $\mu_4 = 0.6$). If two or more users access the same channel, a collision occurs and all the collided users receive zero reward. The percentage of times that the user SU_k accesses successfully his dedicated channel up to n using our algorithm APL is defined as follows:

$$P_k(n) = \frac{1}{n}\sum_{t=1}^{n} 1_{(\text{if } \beta^l_{APL}(t)=k)} \tag{4}$$

where $\beta^l_{APL}(t)$ represents the channel selected at instant t under APL using one of the learning algorithm l such as: TS, UCB or $\epsilon - greedy$. Fig. 2 depicts the $P_k(n)$ of APL under the three learning algorithms. This figure shows clearly that the users converge to their dedicated channels using APL: the first priority user SU_1 converges to the best channel, followed by SU_2, SU_3 and SU_4 respectively. In addition, we can see a fast converges under TS and a slow one under $\epsilon - greedy$. In Fig. 3, we display the cumulated regret of APL under three learning algorithms where a better performance is achieved under TS algorithm compared to UCB and $\epsilon - greedy$. Fig. 3 shows that the regret of APL is logarithmic under the three learning algorithms. Fig. 4 compares the performance of APL, SLK [8] and Musical Chair [10]. As, SLK is based on the UCB algorithm, it can be used just under UCB. While, the Musical Chair is based on random access to estimate the channels availability. However, APL can be used with any learning algorithm. In Fig. 4, APL under TS achieves the lower regret compared to SLK and the Musical Chair where the regret is logarithmic. The Musical Chair produces a constant regret after a finite number of iterations. However, after the exploration phase, the users exploit just the U best channels where the exploration-exploitation phases are separated. Moreover, the priority access cannot be considered in this algorithm and the target of user is to access a random channel among the U best channels. After a large number of slots, the regret of APL under TS can exceed the regret of Musical Chair that is because our algorithm exploits the U best channels, and at the same time surveys /explores the states of others. However, Musical Chair exploits only the U best channels after the exploration phase, this explain why the regret of Musical Chair is constant after a finite time slot. Fig. 5 shows the performance of APL and DMC (Dynamic Musical Chair) for the dynamic access in which the dotted line indicates the entering and leaving of users on the network. Figures (5a) and (5b) represent respectively the cumulated regret and average regret of APL, where at each entering or leaving of users, the regret increases quickly. It is worth

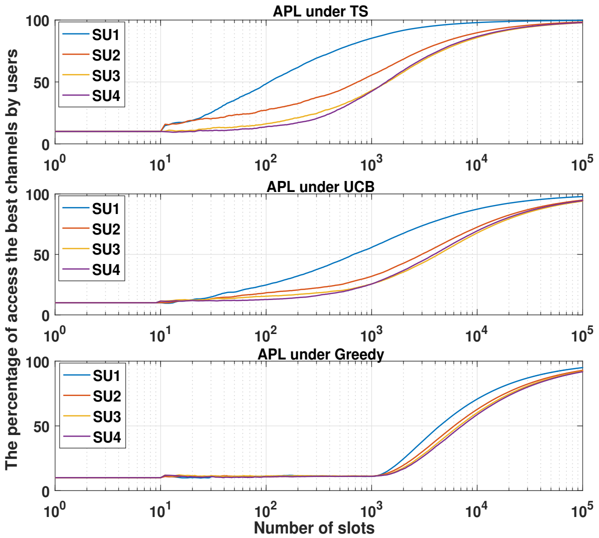

Fig. 2. The percentage of times where each SU_k selects his optimal channel using the proposed approach

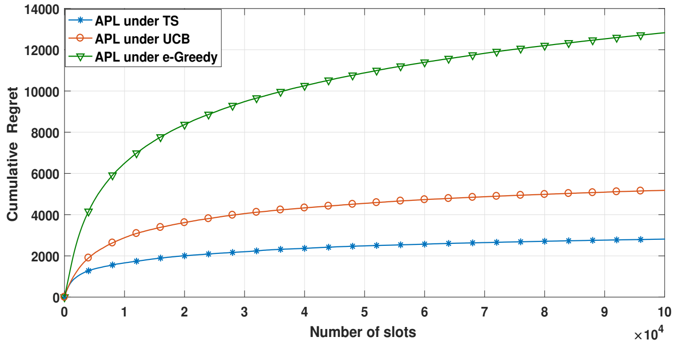

Fig. 3. The regret of APL under TS, UCB, $e - greedy$

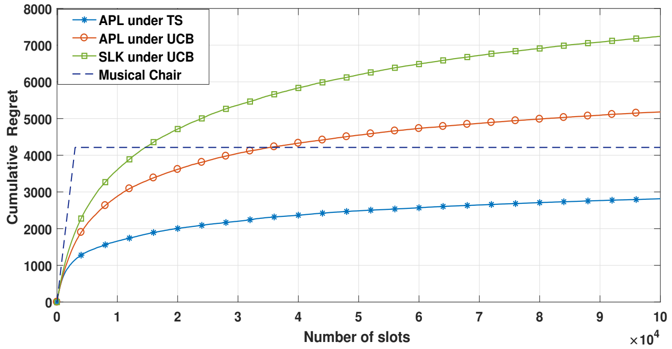

Fig. 4. The regret of APL compared to SLK and Musical Chair

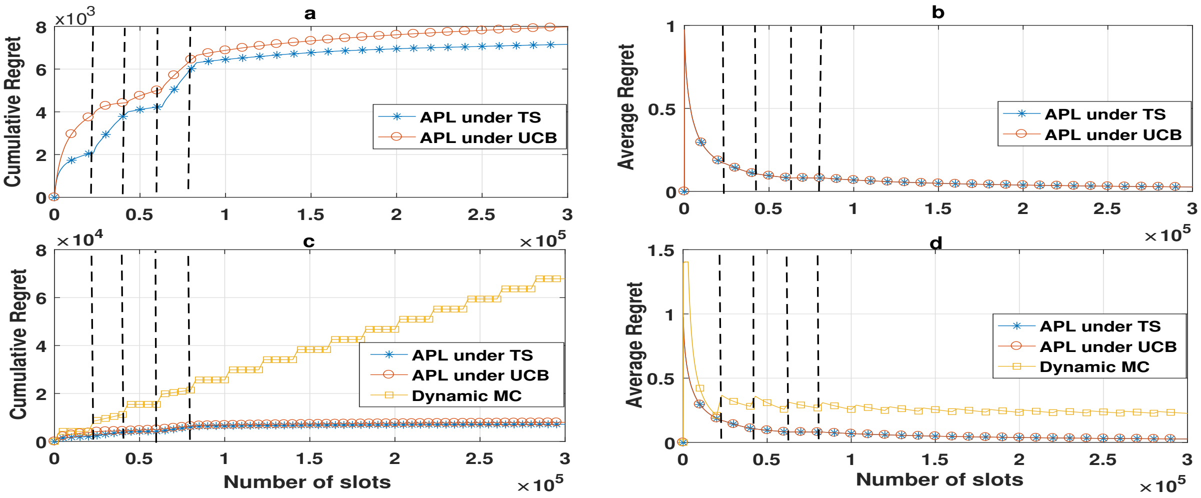

Fig. 5. APL and DMC for dynamic access

mentioning that, in the dynamic scenario and based on our algorithm, the user can change his current channel for two reasons:

- When a collision occurs, SU_k should generate a random rank of the set $\{1, ..., k\}$.
- When a PU accesses the current channel of SU_k, then the index of this channel decreases, and an index of other one exceed this channel.

To the best of our knowledge, two algorithms in the literature that consider the dynamic access: Dynamic Musical Chair [10] and MEGA [11] without consider the priority access. The authors of [10] shows that the Dynamic Musical Chair achieves a better result compared to MEGA algorithm. In Figures (5c) and (5d), we illustrate that our algorithm outperforms the Dynamic Musical Chair and achieves a lower regret. However, after the dynamic access interval, our algorithm achieves a logarithmic regret despite the regret of DMC keeps growing with time. Thus, the access under DMC algorithm is realized in epochs where each one is composed of a learning phase with enough rounds of random exploration to learn the U best channels and the number of users under the dynamic access. The length of each epoch and the learning phase are T_1 and T_0 respectively where these two parameters depend of the number of channels K and the total number of iterations n.

V. Conclusion

In this manuscript, we investigate the problem of OSA in the CR where a novel learning algorithm called APL has been proposed for the scenario of multi-secondary users. This algorithm takes into account the priority dynamic access while only the priority access or the dynamic access are considered in several algorithms such as SLK, $kth - MAB$, DMC or MEGA. Our approach allows for an unknown and variable number of secondary users to access the network where users dynamically enter and leave the system. It is worth noting that this algorithm achieves good experimental results for both fixed and dynamic numbers of users in the network regarding to the global regret. Moreover, APL does not require prior information or cooperation among users as do several existing algorithms. In the future work, we investigate the theoretical analysis of the APL, deriving concrete regret bounds for it.

References

[1] J. Mitola and G. Maguire, “Cognitive radio: making software radios more personal,” *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

[2] A. Nasser, A. Mansour, K. C. Yao, H. Charara, and M. Chaitou, “Spatial and time diversities for canonical correlation significance test in spectrum sensing,” in *European Signal Processing Conference*, Budapest, Hungary, September 2016.

[3] M. Almasri, A. Mansour, C. Moy, A. Assoum, C. Osswald, and D. Lejeune, “Opportunistic spectrum access in cognitive radio for tactical network,” in *European Conference on Electrical Engineering and Computer Science*, Bern, Switzerland, December 2018.

[4] W. Thompson, “On the likelihood that one unknown probability exceeds another in view of the evidence of two samples,” *Biometrika*, vol. 25, no. 3, pp. 285–294, 1933.

[5] T. Lai and H. Robbins, “Asymptotically efficient adaptive allocation rules,” *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[6] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multiarmed bandit problem,” *Machine Learning*, vol. 47, no. 2, pp. 235–256, 2002.

[7] W. Jouini, D. Ernst, C. Moy, and J. Palicot, “Upper confidence bound based decision making strategies and dynamic spectrum access,” in *International Conference on Communications, ICC’10*, Cape Town, South Africa, May 2010.

[8] Y. Gai and B. Krishnamachari, “Decentralized online learning algorithms for opportunistic spectrum access,” in *IEEE Global Communications Conference*, Texas, USA, December 2011.

[9] N. Torabi, K. Rostamzadeh, and V. C. Leung, “Rank-optimal channel selection strategy in cognitive networks,” in *IEEE Global Communications Conference*, California, USA, December 2012.

[10] J. Rosenski, O. Shamir, and L. Szlak, “Multi-player bandits-a musical chairs approach,” in *International Conference on Machine Learning*, New York, USA, June 2016.

[11] O. Avner and S. Mannor, “Concurrent bandit and cognitive radio networks,” in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Nancy, France, September 2014.

[12] R. Kumar, S. J. Darak, A. Yadav, A. K. Sharma, and R. K. Tripathi, “Channel selection for secondary users in decentralized network of unknown size,” *IEEE Communications Letters*, vol. 21, no. 10, pp. 2186–2189, 2017.