Fast Online Learning of Antijamming and Jamming Strategies

Youngjune Gwon
MIT Lincoln Laboratory
gyj@ll.mit.edu

Siamak Dastangoo
MIT Lincoln Laboratory
sia@ll.mit.edu

Carl Fossa
MIT Lincoln Laboratory
cfossa@ll.mit.edu

H. T. Kung
Harvard University
kung@harvard.edu

***Abstract*—Competing Cognitive Radio Network (CCRN) coalesces communicator (comm) nodes and jammers to achieve maximal networking efficiency in the presence of adversarial threats. We have previously developed two contrasting approaches for CCRN based on multi-armed bandit (MAB) and Q-learning. Despite their differences, both approaches have shown to achieve optimal throughput performance. This paper addresses a harder class of problems where channel rewards are time-varying such that learning based on stochastic assumptions cannot guarantee the optimal performance. This new problem is important because an intelligent adversary will likely introduce dynamic changepoints, which can make our previous approaches ineffective. We propose a new, faster learning algorithm using online convex programming that is computationally simpler and stateless. According to our empirical results, the new algorithm can almost instantly find an optimal strategy that achieves the best steady-state channel rewards.**

I. Introduction

Cognitive radios have emerged as a new means to alleviate the spectrum shortage problem. Spectrum is the scarcest (hence, most expensive) resource to build a wireless network, and significant research has focused on improving spectral efficiency and the utility of static allocation methods. In dynamic spectrum access (DSA), an unlicensed or the secondary user is granted an opportunistic access of a licensed spectrum, provided that the user has a proper sensing mechanism to detect the licensees of the channel (*i.e.*, the primary users) and yield discreetly. Generally speaking, cognitive radio research has largely centered around DSA and its commercial aspects.

This paper addresses tactical networking aspects of cognitive radios. In particular, we extend the decision-theoretic framework of Competing Cognitive Radio Network (CCRN) [1], [2] for online learning. We develop a new, fast learning algorithm based on gradient descent that further enhances the performance of cognitive comm and jamming nodes operating under heightened adversarial conditions. The new algorithm aims for faster convergence to optimal antijamming and jamming strategies under dynamic changepoints introduced by an intelligent adversary.

Throughout the paper, we use two hypothetical tactical networks, namely Blue Force Network (BFN or the *ally*) and Red Force Network (RFN or the *enemy*). They clash in a competition to dominate the access to an open spectrum. Differentiated from previous work, RFN can now introduce dynamic changepoints to its channel access and jamming strategies. Subsequently, BFN must address this new challenge where stochastic assumptions on channel reward are no long valid—*i.e.*, channel reward is time-varying. Computing a strategy from reward sampling as in multi-armed bandit (MAB) approaches could suffer from either being too reactive (slow) or having no convergence at all.

Online convex programming [3], [4] motivates the new approach taken in this paper. We first revise the CCRN regret model from the reward-based to a loss version, which allows us to weigh in adversarial viewpoint. This works as if RFN were choosing a loss function for BFN depending on the channel reward performance and sensing BFN node actions. We propose a fast online learning method from computing the gradient of loss function at each horizon. The BFN loss function, however, is not convex, and we cannot straightforwardly apply online convex programming. Therefore, we will propose a new algorithm that addresses such nonconvexity.

The rest of the paper is organized as follows. In Section II, we discuss related work and provide the context of this work. Section III reviews CCRN. Section IV presents a revised mathematical framework for CCRN under dynamic, time-varying adversarial strategy. Section V explains the intuition behind online convex optimization and its applicability for the nonstochastic assumptions of our new problem. We propose a new algorithm, namely CCRN online gradient descent learning. In Section VI, we evaluate our new method and compare its performance to the two previous methods based on MAB and reinforcement Q-learning in a numerical simulation. Section VII concludes the paper.

II. Related Work

This paper extends Competing Cognitive Radio Network (CCRN) by introducing nonstochastic elements. The stochastic multi-armed bandit (MAB) is the basis for one of our previous approaches [1]. In 1933, Thompson [5] introduced a sequential decision problem, later known as stochastic MAB, and proposed a heuristic called Thompson sampling that remained an effective strategy to date. In Bellman 1954 [6], MAB problems were formulated as a class of Markov decision process (MDP). Gittins 1979 [7] proved the existence of a Bayes optimal indexing scheme for MAB problems. Lai & Robbins 1985 [8] introduced the notion of regret, derived its lower bound using the Kullback-Leibler divergence, and constructed asymptotically optimal allocation rules. Anantharam *et al.* 1987 [9] extended Lai & Robbins for multi-player setting. Whittle 1988

This work is sponsored by the Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

[10] introduced PSPACE-hard restless MAB problems and showed that suboptimal indexing schemes are possible. Rivest & Yin 1994 [11] proposed Z-heuristic that achieved a better empirical performance than Lai & Robbins. Auer *et al.* 2002 [12] proposed Upper Confidence Bound (UCB), an optimistic indexing scheme.

Another of our previous approaches [2] models a stochastic Markov game [13] and searches for an optimal solution with reinforcement learning [14]. In particular, Minimax-Q [15], Nash-Q [16], and Friend-or-foe Q (FFQ) [17] provide viable options in decision making whether the competition can be modeled as zero-sum or general-sum games having centralized or distributed controls. This paper also considers similar problems in tactical networking such as Wang *et al.* [18]. They have formulated a stochastic antijamming game played between the secondary user and a malicious jammer, provided sound analytical models, and applied unmodified Minimax-Q learning to solve for the optimal antijamming strategy. Q-learning approaches for CCRN in general have better convergence properties than the MABs. However, the computational complexity of Q-learning could be a practical bottleneck.

III. Competing Cognitive Radio Network (CCRN)

This section provides a brief background on Competing Cognitive Radio Network (CCRN). A CCRN features two types of nodes, communicator (comm) and jammer. Channel accessing by a comm node is determined by sensing vacant spectrum blocks. Jamming an opposing comm node similarly relies on cognition. Spectrum is viewed as being partitioned in time and frequency. There are N non-overlapping channels located at the center frequency f_i (MHz) with bandwidth B_i (Hz) $\forall i = 1, \ldots, N$. A transmission (Tx) opportunity is defined by tuple $\langle f_i, B_i, t, T \rangle$ designating a time-frequency slot at channel i and time t with duration T (msec) as depicted in Fig. 1.

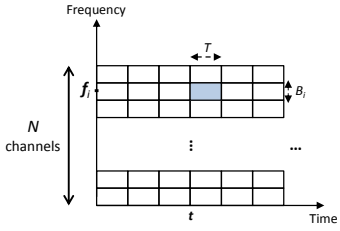

Fig. 1. Tx opportunity $\langle f_i, B_i, t, T \rangle$ (shaded region) in open spectrum access

1) System: The CCRN system consists of sensing, strategy, schedule, and Tx/jam components as illustrated in Fig. 2. We depict two systems Blue Force (BFN) and Red Force (RFN) networks. Using local and global sensing information, a CCRN node applies a strategy to compute an action (*i.e.*, Tx, jam, or do nothing) particular to its channel of interest. The action is scheduled to fill in an opportunity by the system. Node actions can be computed in a centralized or distributed manner.

Under the centralized control, CCRN works as follows.

1) Sense channel activities (each node)

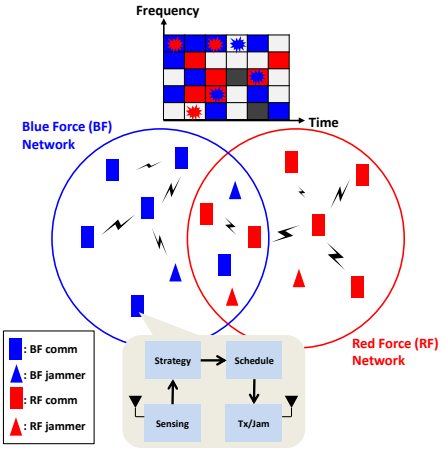

Fig. 2. Competing Cognitive Radio Network (CCRN) systems

2) Collect sensing information (controller)
3) Compute node actions (controller)
4) Disseminate node actions (controller)
5) Act on channel (each node)

In the distributed control, CCRN works as follows.

1) Sense channel activities (each node)
2) Exchange sensing information (each node)
3) Compute its own action (each node)
4) Act on channel (each node)

2) Strategy: A CCRN strategy is the set of rules to select its node actions. A rational strategy *coordinates* to make no conflicting channel access among the nodes. We assume that the nodes exchange control messages. In particular, we follow the approach by Wang *et al.* [18] that assigns control and data channels dynamically. When CCRN finds all of its control channels blocked (*e.g.*, due to jamming) at time t, the spectrum access at $t+1$ will be *uncoordinated*.

3) Reward: A CCRN employs a reward metric to evaluate its strategy. We measure a reward in *bits*. When a comm node makes successful transmission of a packet containing B bits of data, it receives the reward of B (bits). A successful transmission is where only one comm node transmits for an opportunity. If there were two or more, a collision occurs, and no comm node gets a reward. Jammers receive a reward by suppressing an opposing comm node's otherwise successful transmission. A jammer earns a reward B by jamming the slot that an opponent comm node transmits B bits. We call *misjamming* when a jammer jams its own network's comm node (*e.g.*, due to faulty intra-network coordination). Table I summarizes how channel reward is determined.

IV. Mathematical Formulation

A. Notation

CCRN node actions are represented in a vector. At time t, the BFN and RFN actions are $a_B^t = \{a_{B,comm}^t, a_{B,jam}^t\}$ and $a_R^t = \{a_{R,comm}^t, a_{R,jam}^t\}$ for $a_B^t \in A_B$ and $a_R^t \in A_R$, where A_B and A_R are BFN and RFN action sets. Each CCRN action contains both comm and jamming actions. An ith element in

TABLE I
NODE ACTIONS, OUTCOME AND RESULTING REWARD

BF comm	BF jammer	RF comm	RF jammer	Outcome	Reward
Tx	∅	∅	∅	BF Tx success	$R_B \mathrel{+}= B$
∅	Jam	Tx	∅	BF jamming	$R_B \mathrel{+}= B$
Tx	Jam	∅	∅	BF misjamming	–
∅	∅	Tx	∅	RF Tx success	$R_R \mathrel{+}= B$
Tx	∅	∅	Jam	RF jamming	$R_R \mathrel{+}= B$
∅	∅	Tx	Jam	RF misjamming	–
Tx	∅	Tx	∅	Tx collision	–

vector $a^t_{B,comm}$ designates the channel number that the ith BFN comm node tries to transmit at t. Similarly, a jth element in $a^t_{B,jam}$ is the channel that the jth BFN jammer tries to jam at t. The CCRN outcome is $\Omega : A_B \times A_R \rightarrow \mathbb{R}^N$. We map the outcome to a reward $R : \Omega \rightarrow \mathbb{R}$.

B. CCRN Multi-armed Bandit (MAB) Formulation

Multi-armed bandit (MAB) is best explained with a gambler facing N slot machines (arms). The gambler wishes to find a strategy that maximizes $R^t = \sum_{j=1}^{t} r^j$, the *cumulative* reward over a finite horizon t. Lai & Robbins [8] introduced the concept of *regret* for a strategy σ

$$\Gamma^t = t\mu^* - \mathbb{E}\left[R^t_\sigma\right] \tag{1}$$

where μ^* is the hypothetical, maximum average reward if gambler's action were best possible each round. Under σ, the actual reward turns out R^t_σ. Minimizing Γ^t is known mathematically more convenient than maximizing $\mathbb{E}\left[R^t_\sigma\right]$.

For CCRN, an arm is one of channels in the spectrum. Comm nodes and jammers are the players that place Tx and jamming actions to the channels. Since CCRN has multiple nodes, it is a multi-player MAB [9] problem. The BFN strategy σ^t_B is a function over time. For centralized, we write

$$\{x^j_B\}^t_{j=1}, \{a^j_B, \Omega^j\}^{t-1}_{j=1} \xrightarrow{\sigma^t_B} a^t_B \tag{2}$$

where x^t_B is the BFN sensing results at t. For distributed, each BFN node makes own decision

$$x^t_{B,i}, \{x^j_B, a^j_B, \Omega^j\}^{t-1}_{j=1} \xrightarrow{\sigma^t_{B,i}} a^t_{B,i} \tag{3}$$

where $x^t_{B,i}$ is the sensing information only available to BFN node i at time t, and $\sigma^t_{B,i}$ the BFN node i's own strategy.

Thompson sampling [5] is known to provide an optimal performance for stochastic MAB problems. We use Thompson sampling in a Bayesian setup to formulate our MAB-based algorithm for CCRN presented in Algorithm 1 [1]. The algorithm performs the posterior update based on the conjugate prior relationship—*i.e.*, the prior and posterior distributions are the same family of function given the reward's likelihood. Because an optimal strategy should result in the *maximum* channel reward, we consider an extreme-valued likelihood for the CCRN reward. Note that the CCRN reward should be finite. According to extreme value theory [19], the *Weibull* likelihood with inverse gamma prior is the only finite-bound distribution that leads to the rationale behind Algorithm 1. The inverse gamma distribution has two hyperparameters $a, b > 0$. We draw the scale parameter θ from the inverse gamma prior $p(\theta|a,b) = \frac{b^{a-1}e^{-b/\theta}}{\Gamma(a-1)\theta^a}$ for $\theta > 0$ where a and b are the sample mean and variance of the reward of a channel, and $\Gamma(.)$ the gamma function (not to be confused with the Lai & Robbins's regret Γ in Eq. (1)). Then, we sample a Weibull reward using θ drawn from the prior as the reward estimate for the channel. The posterior update follows after the actual reward is learned.

Algorithm 1 (CCRN MAB)

Require: $a_i, b_i = 0\ \forall i$
while $t < 1$ ▷ initialized offline
  Access each channel until $a_i, b_i \neq 0\ \forall i$, where a_i and b_i are sample reward mean and variance
end
while $t \geq 1$ ▷ online
  Draw $\theta_i \sim$ inv-gamma(a_i,b_i)
  Estimate $\hat{r}_i =$ weibull(θ_i,β_i) $\forall i$ for given $0.5 \leq \beta_i \leq 1$
  Access channel $i^* = \arg\max_i \hat{r}_i$
  Observe actual $r^t_{i^*}$ to update $\{R^t_{i^*}, T^t_{i^*}\}$
  Update $a_{i^*} = a_{i^*} + T^t_{i^*}$, $b_{i^*} = b_{i^*} + \sum_t (r^t_{i^*})^{\beta_{i^*}}$
end

C. CCRN Reinforcement Learning Formulation

The Markov game framework [13] can also be used to compute an optimal CCRN strategy. Tuple $\langle S, A_B, A_R, R, T\rangle$ describe the CCRN Markov game between BFN and RFN, where S is the state set, and $A_B = \{A_{B,comm}, A_{B,jam}\}, A_R = \{A_{R,comm}, A_{R,jam}\}$ are the action sets. The reward function $R : S \times \prod A_{\{B,R\},\{comm,jam\}} \rightarrow \mathbb{R}$ maps node actions to a real-valued reward at a given state. The state transition $T : S \times \prod A_{\{B,R\},\{comm,jam\}} \rightarrow \mathrm{PD}(S)$ is the probability distribution over S. A CCRN strategy means the probability distribution over the action set $\pi : S \rightarrow \mathrm{PD}(A)$.

We use reinforcement Q-learning [20] to compute an optimal strategy π^* for CCRN. In particular, we employ the *value iteration* technique that performs an update $Q(s,a) = R(s,a) + \gamma V(s')$ instead of the Bellman equations [21] that optimize the CCRN Markov game in

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} p(s'|s,a) V(s') \tag{4}$$

$$V(s) = \max_{a'} Q(s,a') \tag{5}$$

where s' and a' are the next state and action. Key advantage of Q-learning is to avoid explicit evaluation of the transition probability $p(s'|s,a)$, which is intractable. By linear programming, we can compute optimal $\pi^* = \arg\max_\pi \sum_a Q(s,a)\,\pi$ subject to the value maximization. In Algorithm 2, we present the Minimax-Q learning algorithm for CCRN [2]. We remark that there are other Q-learning algorithms plausible for CCRN such as Nash-Q and Friend-or-foe Q.

D. New Formulation under Time-varying Channel Reward

In stochastic setting, the bottomline for learning a strategy is to estimate unknown reward distribution $\mathcal{R}_{a_B,a_R} = \mathbb{P}\left[r|a_B, a_R\right]$. Presumably, if we have accurate sensing capability, we can learn stable estimate of the distribution over time.

Algorithm 2 (CCRN Q-learning)

Require: $Q(s, a_B, a_R) = 1, V(s) = 1, \pi(s, a_B) = \frac{1}{|\mathcal{A}|}$ $\forall$ state $s \in \mathcal{S}$, BF action $a_B \in \mathcal{A}$, RF action $a_R \in \mathcal{A}$; learning rate $\alpha < 1$ with decay $\lambda \leq 1$ (α, λ nonnegative)

while $t \geq 1$
  Draw $a_B^t \sim \pi(s^t)$ and execute
  Observe r_B^t
  Estimate a_R^t given observed reward
  Compute s^{t+1}
  $Q(s^t, a_B^t, a_R^t) = (1-\alpha)Q(s^t, a_B^t, a_R^t) + \alpha(r_B^t + \gamma V(s^{t+1}))$
  linprog: $\pi(s^t, .) = \arg\max_\pi \sum_{a_B} \pi(s^t, a_B) Q(s^t, a_B, a_R)$
  Update $V(s^t) = \min_{a_R} \sum_{a_B} \pi(s^t, a_B) Q(s^t, a_B, a_R)$
  Update $\alpha = \lambda \times \alpha$
end

The optimal regret bound for stochastic MAB is well-studied and known as $O(\log T)$.

Auer *et al.* [22] provides some useful background for *nonstochastic* MAB suitable for our new scenario. Their adversarial assumptions include rewards deliberately altered by the opponent. This is possible when the BFN faces an intelligent RFN that has matched cognitive abilities and can learn as effectively as BFN.

In *adversarial* bandits, we revise the classical Lai & Robbins regret using some loss function $l^t(.)$:

$$\Upsilon^T = \sum_{t=1}^{T} l^t(a_B^t) - \min_{a_B^{t*} \in A_B} \sum_{t=1}^{T} l^t(a_B^{t*}) \tag{6}$$

The gain (*i.e.*, with reward) and loss versions of the regret are symmetric. The intuition behind the loss version is that we want an adversarial view as if the RF network were choosing $l^t(.)$ in the beginning of t and reveals only the quantity $l^t(a_B^t)$ upon the BF placing its action a_B^t. Note that $l^t(.)$ evolves over time as it is a function of time. In the next section, we use this revised regret, which has adversarial point of view, to devise a faster, online learning algorithm.

V. Finding Optimal Actions with Online Learning

This section presents a new algorithm to compute the joint antijamming and jamming actions for CCRN. The new method is based on gradient descent and requires no offline training.

A. Online Convex Optimization

Imagine that RFN (the adversary) chooses its loss function $l^t(.)$ at time t from a hidden sequence $l^1, l^2, l^3, \ldots$ of *convex* functions. BFN chooses its action a_B^t also from some convex set $\mathcal{K} \subseteq \mathbb{R}^N$ for $t = 1, \ldots, T$. For clarity, let $\max_{a_B^t \in \mathcal{K}} l^t(a_B^t) \leq 1$. Can the regret in Eq. (6) grow sublinearly with respect to T?

For this setup, Flaxman *et al.* [4] propose a simple gradient approximation. The gradient can be computed from evaluating $l^t(.)$ at a single random point. Despite such bias, they show that the resulting gradient estimate is sufficient to achieve a regret bound of $O(T^{3/4})$. The key to their solution is *online convex programming* developed by Zinkevich [3].

Online convex programming finds a point in a convex set $F \subseteq \mathbb{R}^N$ that minimizes a convex cost function $c : F \to \mathbb{R}$. If the convex set F is known, online convex programming will result in the cost bound of $O(\sqrt{T})$ for a total of T rounds. Algorithm 3 presents GIGA (Generalized Infinitesimal Gradient Ascent), a template for the online gradient descent.

Algorithm 3 (GIGA)

while $t \geq 1$
  play action $a^t \in \mathcal{K}$
  observe regret $l^t(a^t)$
  compute estimate $\hat{g}^t$ of loss gradient $\nabla l^t(a^t)$
  $\Delta^{t+1} := a^t - \eta\, \hat{g}^t$
  $a^{t+1} := \arg\min_{a \in \mathcal{K}} \left\| a - \Delta^{t+1} \right\|$
end

The approach by Flaxman *et al.* [4] is essentially a GIGA with the gradient estimate

$$\hat{g}^t = \frac{N}{\delta} l^t(a^t + \delta\, u)\, u \tag{7}$$

where N denotes dimensionality of the action space (*i.e.*, $a \in \mathcal{K} \subseteq \mathbb{R}^N$), u a random unit vector, and some small $\delta > 0$.

B. New Algorithm

We propose Algorithm 4 based on online gradient descent learning. Straightforward adoption of GIGA (Algorithm 3) for CCRN is problematic for two reasons. First, the loss function for CCRN is not convex. It is likely a mixture of convex and concave curves as depicted in Fig. 3. Hence, an unmodified gradient descent method such as GIGA will result in a vastly different outcome depending the initial point. For example, if the initial action were a_1, the gradient descent would take it to $l_1^* = l^t(a_1^*)$, a local minimum loss close to $l^t(a_1)$. Note that a_1^* is the corresponding optimal action computed iteratively from a_1 by descending the gradient of loss. If the initial action were a_2, we would achieve l_2^* as illustrated in Fig. 3.

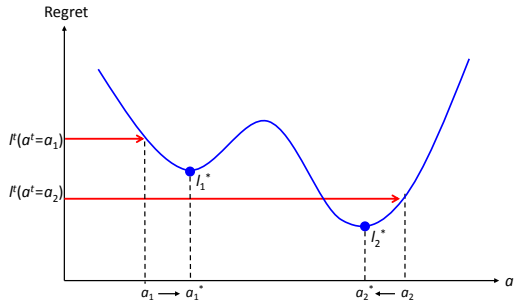

Fig. 3. Gradient descent for CCRN is problematic.

Accurate loss function estimation gives another issue to apply gradient descent in CCRN. We expect to learn the loss function from sensing results collected from multiple CCRN nodes. If there are too many channels to learn compared to the number of CCRN nodes (*i.e.*, $N \gg M$), our learning suffers severely from partial feedback assuming that the CCRN sensing capacity as a whole is proportional to the number of nodes M.

We now explain key principles of Algorithm 4.

- **Initialize to random action.** Given no offline training or prior knowledge, the new algorithm starts at random.
- **Estimate loss function from observed regret.** The BFN loss function is a function of RFN node actions, consisting of multiple convex and concave regions. Given BFN node actions, the BFN comm and jamming loss functions are derived from sensing results that estimate a_{RC} and a_{RJ}, RFN comm and jamming actions:

$$l_{BC} = \|a_{BC}\|_0 - a_{BC} \cdot \neg(a_{RC} \vee a_{RJ})$$
$$l_{BJ} = \|a_{BJ}\|_0 - a_{BJ} \cdot (a_{RC} \vee a_{RJ})$$

- **Compute gradient.** From the BFN action space, the algorithm searches for a_+ and a_- that differ from the current action a by the smallest (*e.g.*, one bit) possible. The gradient is then computed using the estimated loss functions l_{BC} and l_{BJ} with a_+ and a_-.
- **Choose new action.** The estimated gradient of the loss function serves the guidance whether or not the current action has to sustain or change. The loss estimates at a_+ and a_- are better than that of a, the algorithm chooses the better of a_+ and a_-. If a is at one of the undesirable local minima, the final **else** clause of Algorithm 4 is executed to escape the region around a for better.

Algorithm 4 (CCRN online gradient descent learning)

choose a^1 randomly
while $t \geq 1$
  execute a^t and observe r^t
  compute $\hat{l}^t(a^t)$
  if $|l^* - \hat{l}^t(a^t)| < \epsilon$
    $a^{t+1} := a^t$
    continue
  end
  $a^t_- := a^t - \delta_-$ such that $\|a^t\|_0 = \left\|a^t_-\right\|_0$
  $a^t_+ := a^t + \delta_+$ such that $\|a^t\|_0 = \left\|a^t_+\right\|_0$
  $\nabla\hat{l}^t := \min\{\hat{l}^t(a^t_-), \hat{l}^t(a^t_+)\}$
  if $\nabla\hat{l}^t < \hat{l}^t(a^t)$
    $a^{t+1} := \arg\min_{x \in \{a^t_-, a^t_+\}} \hat{l}^t(x)$
  else
    $a^{t+1} := a^t - w + u$
  end
end

VI. Evaluation

We evaluate the performance of Algorithm 4 along Algorithm 1 (stochastic MAB) and Algorithm 2 (Minimax-Q) against Algorithm 5 (benchmark) that describes an adversarial CCRN with random changepoint of strategy.

A. Scenario, Benchmark Algorithm, and Metric

We have implemented a custom MATLAB simulator. We configure BFN to run either Algorithm 1, 2, or 4 while fixing RFN with Algorithm 5. The benchmark algorithm randomly draws RFN node actions and holds for random T time slots. We compare convergence properties of the new algorithm against our old CCRN algorithms against RFN's time-varying strategy embodied in the benchmark algorithm. We also examine the reward performance of BFN using average reward per channel as the evaluation metric

$$\bar{R}^t = \frac{1}{M \cdot t} \sum_{j=1}^{t} \sum_{i=1}^{N} r_i^j$$

where r_i^j is the ith channel reward at $t = j$, and there are M nodes in the CCRN trying out N channels in the spectrum. To determine r_i, we apply all available sensing results to the decision matrix of Table I. Using $B = 1$ (normalized bit reward) yields the following:

- $r_i^t = 1$ if only one comm node transmits and no jamming in channel i at t;
- $r_i^t = 1$ if a jammer jams the sole opposing comm's transmission in channel i at t;
- $r_i^t = 0$ otherwise.

Algorithm 5 (Random changepoint of strategy)

while $t \geq 1$
  draw random $a \in A$
  choose T randomly
  for T slots
    play action a
  end
end

We have simulated a spectrum with $N = 10$, 20, 30, 40, and 50 channels. We have also varied the total number of nodes M from 10 to 50. For $M = 10$, we have placed $J = 2$ jammers per each network (hence, the number of comm nodes $C = M - J = 8$). We grow 2 jammers per additional 10 nodes. That is, we set $J = 4$ for $M = 20$, $J = 6$ for $M = 30$, $J = 8$ for $M = 40$, and $J = 10$ for $M = 50$. Both comm nodes and jammers have a transmit probability $p_{Tx} = 1$ for each time slot. Each simulation runs the total of 5,000 time slots.

B. Discussion of Results

Figure 4 plots the convergence time for each learning method. Note that the convergence time is the number of slots required for BFN to establish a steady-state reward. Such equilibrium is at least maintained until the next changepoint introduced by RFN that chooses random node actions. The plot shows convergence times for each BFN strategy resulted from all possible values of N and M used in the evaluation. The new algorithm based online learning shows the best convergence property with drastically flatter curve (*i.e.*, faster time to steady-state) than the other two algorithms.

In Figure 5, we highlight average cumulative reward for BFN under $N = 40$ and $M = 20$. We observe very similar steady-state reward performances from the three different CCRN strategies. This is expected since all three algorithms are capable of achieving the optimal CCRN reward performance. The difference, however, is evident for $t \leq 500$ slots.

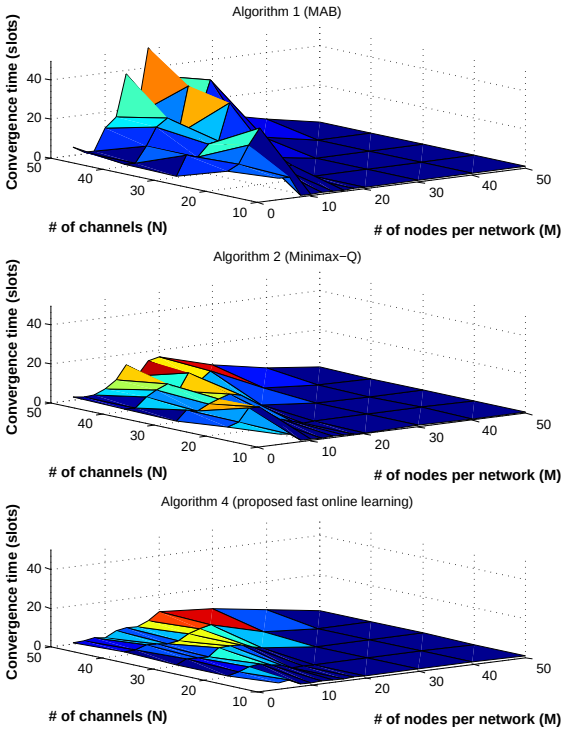

Fig. 4. Convergence time comparison

The proposed algorithm is much faster to find optimal BFN actions under multiple, random changepoints for RFN strategy.

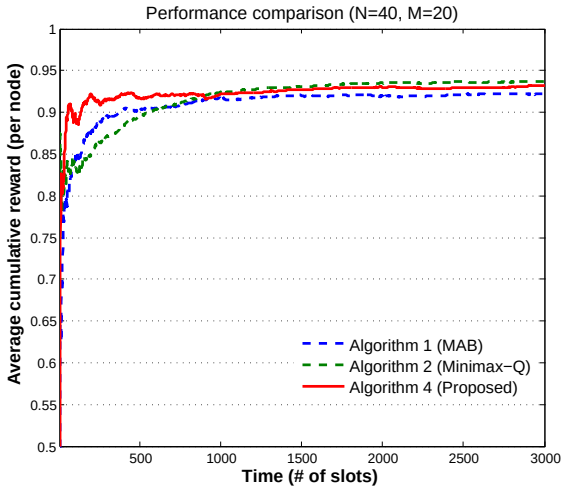

Fig. 5. Reward performance comparison

VII. Conclusion

We have addressed a harder class of problems in determining optimal media access strategies for Competing Cognitive Radio Network (CCRN). Differentiated from previous work, we consider nonstochastic, time-varying channel rewards caused by an intelligent adversary, another CCRN capable of making sound antijamming and jamming strategies. To cope with dynamic changepoints induced by the adversary, we require a new CCRN strategy that has better convergence properties. We have proposed a fast online learning algorithm for CCRN. The new algorithm is based on gradient descent, requires estimates from unacted channels, but is computationally simpler and stateless. According to our empirical benchmark, the new algorithm can almost instantly find an optimal strategy that achieves the best steady-state reward. The new algorithm can be further improved by the use of myopic channel activity predictors. We plan to improve our work with channel activity classifiers and predictors built on machine learning.

References

[1] Y. Gwon, S. Dastangoo, and H. Kung, "Optimizing Media Access Strategy for Competing Cognitive Radio Networks," in *IEEE GLOBECOM*, 2013.

[2] Y. Gwon, S. Dastangoo, C. Fossa, and H. Kung, "Competing Mobile Network Game: Embracing Antijamming and Jamming Strategies with Reinforcement Learning," in *IEEE Communications and Network Security (CNS)*, 2013.

[3] M. Zinkevich, "Online Convex Programming and Generalized Infinitesimal Gradient Ascent," in *ICML*, 2003.

[4] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online Convex Optimization in the Bandit Setting: Gradient Descent Without a Gradient," in *SODA*, 2005.

[5] W. R. Thompson, "On the Likelihood That One Unknown Probability Exceeds Another in View of the Evidence of Two Samples," *Biometrika*, vol. 25, no. 3-4, pp. 285–294, 1933.

[6] R. Bellman, *A Problem in the Sequential Design of Experiments*. Defense Technical Information Center, 1954.

[7] J. C. Gittins, "Bandit Processes and Dynamic Allocation Indices," *Journal of the Royal Statistical Society*, vol. 41, no. 2, pp. 148–177, 1979.

[8] T. L. Lai and H. Robbins, "Asymptotically Efficient Adaptive Allocation Rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[9] V. Anantharam, P. Varaiya, and J. Walrand, "Asymptotically Efficient Allocation Rules for Multiarmed Bandit Problem with Multiple Plays–Part I: I.I.D. Rewards," *IEEE Trans. on Automatic Control*, vol. 32, no. 11, pp. 968–976, Nov 1987.

[10] P. Whittle, "Restless Bandits: activity allocation in a changing world," *Journal of Applied Probability*, vol. 25A, pp. 287–298, 1988.

[11] R. L. Rivest and Y. Yin, "Simulation Results for a New Two-armed Bandit Heuristic," in *Workshop on Computational Learning Theory and Natural Learning Systems*, 1994.

[12] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time Analysis of the Multiarmed Bandit Problem," *Machine Learning*, vol. 47, no. 2-3, pp. 235–256, May 2002.

[13] L. S. Shapley, "Stochastic Games," *Proc. of the National Academy of Sciences*, 1953.

[14] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[15] M. L. Littman, "Markov Games as a Framework for Multi-agent Reinforcement Learning," in *Proc. of International Conference on Machine Learning (ICML)*, 1994.

[16] J. Hu and M. P. Wellman, "Multiagent Reinforcement Learning: Theoretical Framework and an Algorithm," in *Proc. of the International Conference on Machine Learning (ICML)*, 1998.

[17] M. L. Littman, "Friend-or-foe Q-learning in General-sum Games," in *Proc. of International Conference on Machine Learning (ICML)*, 2001.

[18] B. Wang, Y. Wu, K. Liu, and T. Clancy, "An Anti-jamming Stochastic Game for Cognitive Radio Networks," *IEEE JSAC*, vol. 29, no. 4, 2011.

[19] L. de Haan and A. Ferreira, *Extreme Value Theory: An Introduction*. Springer, 2006.

[20] C. Watkins and P. Dayan, "Q-learning," *Machine Learning*, 1992.

[21] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

[22] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The Nonstochastic Multiarmed Bandit Problem," *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002.