Competing Mobile Network Game: Embracing Antijamming and Jamming Strategies with Reinforcement Learning

Youngjune Gwon
Harvard University
gyj@eecs.harvard.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*—We introduce Competing Mobile Network Game (CMNG), a stochastic game played by cognitive radio networks that compete for dominating an open spectrum access. Differentiated from existing approaches, we incorporate both communicator and jamming nodes to form a network for friendly coalition, integrate antijamming and jamming subgames into a stochastic framework, and apply Q-learning techniques to solve for an optimal channel access strategy. We empirically evaluate our Q-learning based strategies and find that Minimax-Q learning is more suitable for an aggressive environment than Nash-Q while Friend-or-foe Q-learning can provide the best solution under distributed mobile ad hoc networking scenarios in which the centralized control can hardly be available.**

I. Introduction

Cognitive radios have arisen commercially over the last decade, enabling a new means to share radio spectrum. Dynamic spectrum access (DSA) [1] is a compelling usage scenario for the cognitive radio system. DSA aims to relieve shortages of radio spectrum, which is the scarcest—hence, the most expensive—resource to build a wireless network. Much of contemporary research has viewed cognitive radios as the secondary user of a licensed spectrum and focused on the development of a flexible mechanism to opportunistically access the licensed channel to its maximal spectral efficiency.

While cognitive radios are deemed a commercial success, their applicability in tactical wireless networking is even more adequate. The central concept behind the cognitive radio system is intelligent decision making, which makes it suitable for operating in a hostile, competing wireless environment. In this paper, we introduce Competing Mobile Network Game (CMNG) where radio nodes form a tactical wireless network and strategize holistically as a team to best its opponent in dominating the access to an open spectrum. We are particularly interested in leveraging knowledge acquired through sensing and learning to overcome extreme operational characteristics of the radio network such as jamming attacks. Also, our tactical settings embrace jamming as a strategy to suppress communication activities of an opponent.

In an *antijamming* game, the radio network attempts to maximize its communication utility under the presence of hostile jamming devices, whereas its friendly jammers aim to minimize the opposing network's communication in a *jamming* game. Existing game-theoretic frameworks for cognitive radios [2]–[4] have treated the antijamming and jamming problems separately. We depart from the existing approaches and integrate antijamming and jamming games to jointly solve for an optimal strategy, exploring Q-learning techniques used in reinforcement learning. Given an optimistic assumption of perfect sensing at the lower layer that allows correct outcome of a channel to be fed back, Q-learning can result in optimal channel access decisions that lead to the best cumulative average reward in a steady state.

The rest of the paper is organized as follows. In Section II, we explain our system model and underlying assumptions. Section III presents mathematical formulation of CMNG. In Section IV, we apply reinforcement learning to determine optimal strategies for CMNG and show how Q-learning can be used to solve antijamming and jamming games. We propose both centralized and distributed control approaches based on Minimax-Q, Nash-Q, and Friend-or-foe Q-learning algorithms. In Section V, we evaluate the proposed methods with numerical simulation and analyze their performance. In Section VI, we discuss related work and provide the context of our work. Section VII concludes the paper.

II. Model

A. Competing Mobile Networks

For clarity of discussion, let us consider two mobile networks, namely Blue Force (BF) or the *ally* and Red Force (RF) or the *enemy* networks. Each network consists of two types of nodes: communicator (comm) node and jammer. BF and RF networks compete fiercely to achieve higher comm data throughput and prevent the opponent's comm activities by jamming. The primary-secondary user dichotomy popular in the DSA literature is not applicable here, and little or no fixed infrastructural support is assumed. Mobile ad hoc network (MANET) would be the most convincing network model, but the network-wide cooperation and strategic use of jamming against the opponent are critical to design a winning media access scheme. A competing mobile network can adopt a centralized control model where the node actions are coordinated through a singular entity that makes coherent, network-wide decisions. On the other hand, a distributed control model allows each node to decide its own action. We will evaluate both models in later sections of this paper.

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.

B. Communication Model

Spectrum available for open access is partitioned in time and frequency. There are N non-overlapping channels located at the center frequency f_i (MHz) with bandwidth B_i (Hz) for $i = 1, \ldots, N$. A transmission opportunity is represented by a tuple $\langle f_i, B_i, t, T \rangle$, which designates a frequency-time slot at channel i and time t with time duration T (msec) as depicted in Fig. 1. We assume a simple CSMA in which comm nodes first sense before transmitting in a slot of opportunity.

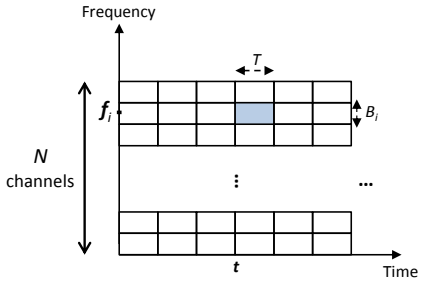

Fig. 1. Transmission opportunity $\langle f_i, B_i, t, T \rangle$ (shaded region)

In order to coordinate a coherent spectrum access and jamming strategy network-wide, we assume that the nodes (both comm and jammers) exchange necessary information via control messages. We call the channels used to exchange control messages 'control channels.' On the contrary, 'data channels' are used to transport regular data packets. We follow the DSA approach [2] that control or data channels are dynamically determined and allocated. When a network finds all of its control channels blocked (*e.g.*, due to jamming) at time t, the spectrum access at time $t+1$ will be uncoordinated.

C. Jamming

Xu *et al.* [5] provides widely accepted taxonomy of jammers. A constant jammer continuously dissipates power into a selected channel by transmitting arbitrary waveforms. A deceptive jammer can instead send junk bits encapsulated in a legitimate packet and conceal its intent to disrupt comm nodes. A random jammer alternates between jamming and remaining quiet for random time intervals. A reactive jammer listens to the channel, stays quiet when the channel is idle, and starts transmitting upon sensing an activity. We add *strategic* jammer into the existing taxonomy, which is similar to the statistical jammer described in Pajic and Mangharam [6]. A strategic jammer, however, is more intelligent—it can adapt to media accessing patterns of comm nodes, learn antijamming schemes, and operate without being detected for long, causing severe damages.

D. Rewards

A comm node receives a reward of B (bits) upon a successful transmission at the attempted slot. Definition of the successful transmission follows the classic ALOHA, which requires that there should be only one comm node transmission per Tx opportunity. If there were two or more simultaneous transmissions at a Tx opportunity (regardless of the same or opposing network comm nodes), a collision would occur, and no comm node gets a reward.

TABLE I
OUTCOME AND RESULTING REWARD AT TX OPPORTUNITY SLOT

BF comm	BF jammer	RF comm	RF jammer	Outcome	Reward
Tx	∅	∅	∅	BF Tx success	$R_{BF} \mathrel{+}= B$
∅	Jam	Tx	∅	BF jamming	$R_{BF} \mathrel{+}= B$
Tx	Jam	∅	∅	BF misjamming	None
∅	∅	Tx	∅	RF Tx success	$R_{RF} \mathrel{+}= B$
Tx	∅	∅	Jam	RF jamming	$R_{RF} \mathrel{+}= B$
∅	∅	Tx	Jam	RF misjamming	None
Tx	∅	Tx	∅	Tx collision	None

Jammers do not create any reward by themselves but can take away an opposing comm node's otherwise successful transmission. For example, a BF jammer earns a reward B by jamming the slot in which a sole RF comm node transmits. If there were no jamming, the RF comm node would have earned B. Also, a BF jammer can jam a BF comm mistakenly (*e.g.*, due to faulty intra-network coordination), which we call *misjamming*. Table I summarizes the outcome at a slot of transmission opportunity ('∅' means no action).

III. MATHEMATICAL FORMULATION OF COMPETING MOBILE NETWORK GAME (CMNG)

This section provides formal introduction of Competing Mobile Network Game (CMNG), a stochastic game for competing cognitive radio networks, Blue Force (BF) and Red Force (RF).

A. Basic Definitions and Objective

We define CMNG the tuple:

$$\mathcal{G}_{\text{CMNG}} = \langle S, A_B, A_R, R, T \rangle$$

where S is the set of states, and $A_B = \{A_{B,comm}, A_{B,jam}\}$, $A_R = \{A_{R,comm}, A_{R,jam}\}$ are the action sets of BF and RF networks. Notice that the action sets break down to include both the comm and jammer actions. CMNG is a stochastic game [7], which extends Markov Decision Process (MDP) [8] by incorporating an agent as the game's policy maker that interacts with an environment possibly containing other agents. Under the centralized control model (Section II.A), CMNG considers one agent per network that computes strategies for all nodes in the network whereas there are multiple agents (*i.e.*, each node) per network under the distributed control model.

We interchangeably use the terms *strategy* and *policy* of the stochastic game $\pi : S \to \text{PD}(A)$ that denotes the probability distribution over the action set. CMNG has the reward function $R : S \times \prod A_{\{B,R\},\{comm,jam\}} \to \mathbb{R}$ that maps node actions at a given state to a reward value. The state transition function $T : S \times \prod A_{\{B,R\},\{comm,jam\}} \to \text{PD}(S)$ is the probability distribution over S. S, A, π, and R evolve over time, thus are functions of time. We use a lower case letter with superscripted t for their realization in time (not tth power of), *e.g.*, s^t ($\in S$) means the CMNG state at time t, and similarly a_B^t ($\in A_B$) and a_R^t ($\in A_R$) for BF and RF node actions at t.

TABLE II
COLLISION PARAMETERS

Parameter	Description
$I_{B,C}$	# of control channel collisions caused by BF comms only
$I_{B,D}$	# of data channel collisions caused by BF comms only
$I_{R,C}$	# of control channel collisions caused by RF comms only
$I_{R,D}$	# of data channel collisions caused by RF comms only
$I_{BR,C}$	# of control channel collisions caused by BF and RF comms
$I_{BR,D}$	# of data channel collisions caused by BF and RF comms

TABLE III
JAMMING PARAMETERS

Parameter	Description
$J_{B,R,C}$	# of BF control channel jammed by RF jammers
$J_{B,R,D}$	# of BF data channel jammed by RF jammers
$J_{B,B,C}$	# of BF control channel jammed by BF jammers
$J_{B,B,D}$	# of BF data channel jammed by BF jammers
$J_{B,BR,C}$	# of BF control channel jammed by BF and RF jammers
$J_{B,BR,D}$	# of BF data channel jammed by BF and RF jammers
$J_{R,B,C}$	# of RF control channel jammed by BF jammers
$J_{R,B,D}$	# of RF data channel jammed by BF jammers
$J_{R,R,C}$	# of RF control channel jammed by RF jammers
$J_{R,R,D}$	# of RF data channel jammed by RF jammers
$J_{R,BR,C}$	# of RF control channel jammed by BF and RF jammers
$J_{R,BR,D}$	# of RF data channel jammed by BF and RF jammers

The objective of CMNG is to win in the competition of dominating the open spectrum access, which can be achieved by transporting or jamming more comm data bits. For BF network, this is equivalent to find an optimal distribution π^* of possible actions that maximizes the expected cumulative sum of discounted rewards:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s^t, a_B^t, a_R^t)\right] \quad (1)$$

where γ is a reward discount ratio, strategy π decides BF node actions, RF node actions are measurable to determine the state, and the reward can be observed over time.

B. States and Actions

Consider that the spectrum under competition is partitioned in N channels, each of which can be described by a Markov chain. If there are L discrete states for each channel, we require to track L^N states for CMNG. Unfortunately, this results in $O(L^N)$, an exponential complexity class with respect to the number of channels. We instead choose a terser state representation $s = \langle I_C, I_D, J_C, J_D\rangle$ where I_C denotes the number of control channels collided, I_D the number of data channels collided, J_C the number of control channels jammed, and J_D the number of data channels jammed.

Given the current state and the action sets of BF and RF nodes, the next state of CMNG is computable. The actions of the opponent is inferred from channel measurements and sensing. To estimate I_C, I_D, J_C, and J_D, we need to observe the parameters in Tables II and III to calculate

$$\begin{aligned} I_C &= \sum_{x\in\{B,R,BR\}} I_{x,C} \\ I_D &= \sum_{x\in\{B,R,BR\}} I_{x,D} \\ J_C &= \sum_{x\in\{B,R\},y\in\{B,R,BR\}} J_{x,y,C} \\ J_D &= \sum_{x\in\{B,R\},y\in\{B,R,BR\}} J_{x,y,D} \end{aligned}$$

For illustrative purposes, we present an example where each BF and RF network has $C = 2$ comm nodes and $J = 2$ jammers, and there are $N = 10$ channels in the spectrum. Suppose the channels are numbered $1, \ldots, 10$. The BF node actions at t are $a_B^t = \langle a_{B,comm}^t, a_{B,jam}^t\rangle$ where $a_{B,comm}^t$ and $a_{B,jam}^t$ are vectors of sizes C and J, and similarly $a_R^t = \langle a_{R,comm}^t, a_{R,jam}^t\rangle$ for the RF node actions. Let $a_{B,comm}^t = [7\ 3]$; this means that BF comm node 1 transmits in channel 7, and BF comm node 2 in channel 3. Let $a_{B,jam}^t = [1\ 5]$; that is, BF jammer 1 jams channel 1, and BF jammer 2 jams channel 5. For RF network, let $a_{R,comm}^t = [3\ 5]$ and $a_{R,jam}^t = [10\ 9]$. Also, BF network uses channel 2 for control, and the RF control channel is channel 1. These node actions and control channel usages form a bitmap shown in Fig. 2 where 1 indicates transmit, jam, or markup as control channel, and 0 otherwise. Both BF jammers are successful here, jamming the RF control and comm data transmissions in channels 1 and 5, respectively. BF and RF comm data transmissions collide in channel 3, and BF has a successful data transmission in channel 7 whereas RF has no success in comm data. RF jammers end up unsuccessfully, jamming empty channels 9 and 10. This example results in state $s^t = \langle I_C = 0, I_D = 1, J_C = 1, J_D = 1\rangle$.

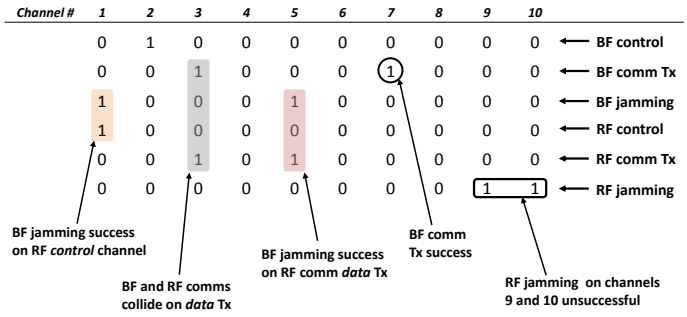

Fig. 2. CMNG action-state computation example

C. State Transition Probability Distribution

In this section, we derive the full, analytical formula for the CMNG state transition probability distribution that can be used for numerical approximation.

1) Counting parameters for state transition: The following conditional probability distribution determines the transition function T:

$$\begin{aligned} &p(s^{t+1}|s^t, a_B^t, a_R^t) \\ &\quad = p(I_C^{t+1}, I_D^{t+1}, J_C^{t+1}, J_D^{t+1}|I_C^t, I_D^t, J_C^t, J_D^t, a_B^t, a_R^t) \end{aligned}$$

To express $I_C^{t+1}, I_D^{t+1}, J_C^{t+1}$, and J_D^{t+1}, we need to define the counting parameters related to collision and jamming:

- $m_{C1} \stackrel{\text{def}}{=}$ # of collided control channels previously uncollided and unjammed;
- $m_{C2} \stackrel{\text{def}}{=}$ # of collided control channels previously collided;
- $m_{C3} \stackrel{\text{def}}{=}$ # of collided control channels previously jammed;
- $m_{D1} \stackrel{\text{def}}{=}$ # of collided data channels previously uncollided and unjammed;
- $m_{D2} \stackrel{\text{def}}{=}$ # of collided data channels previously collided;
- $m_{D3} \stackrel{\text{def}}{=}$ # of collided data channels previously jammed;
- $n_{C1} \stackrel{\text{def}}{=}$ # of jammed control channels previously uncollided and unjammed;
- $n_{C2} \stackrel{\text{def}}{=}$ # of jammed control channels previously collided;
- $n_{C3} \stackrel{\text{def}}{=}$ # of jammed control channels previously jammed;
- $n_{D1} \stackrel{\text{def}}{=}$ # of jammed data channels previously uncollided and unjammed;
- $n_{D2} \stackrel{\text{def}}{=}$ # of jammed data channels previously collided;
- $n_{D3} \stackrel{\text{def}}{=}$ # of jammed data channels previously jammed.

Now we can write the number of collided control channels $I_C^{t+1} = m_{C1} + m_{C2} + m_{C3}$, the total number of collided data channels $I_D^{t+1} = m_{D1} + m_{D2} + m_{D3}$, the jammed control channels $J_C^{t+1} = n_{C1} + n_{C2} + n_{C3}$, and the jammed data channels $J_D^{t+1} = n_{D1} + n_{D2} + n_{D3}$.

We define the counting parameters that describe how BF and RF networks choose control and data channels at time t:

- $\alpha_{C1}^t \stackrel{\text{def}}{=}$ # of control channels chosen from previously uncollided and unjammed channel space;
- $\alpha_{D1}^t \stackrel{\text{def}}{=}$ # of data channels chosen from previously uncollided and unjammed channel space;
- $\alpha_{C2}^t \stackrel{\text{def}}{=}$ # of control channels chosen from previously collided channel space;
- $\alpha_{D2}^t \stackrel{\text{def}}{=}$ # of data channels chosen from previously collided channel space;
- $\alpha_{C3}^t \stackrel{\text{def}}{=}$ # of control channels chosen from previously jammed channel space;
- $\alpha_{D3}^t \stackrel{\text{def}}{=}$ # of data channels chosen from previously jammed channel space.

We define the parameters to describe how BF and RF jamming actions are chosen at t:

- $\alpha_{I1}^t \stackrel{\text{def}}{=}$ # of channels chosen from previously uncollided channel space for jamming;
- $\alpha_{I2}^t \stackrel{\text{def}}{=}$ # of channels chosen from previously collided channel space for jamming;
- $\alpha_{J1}^t \stackrel{\text{def}}{=}$ # of channels chosen from previously unjammed channel space for jamming;
- $\alpha_{J2}^t \stackrel{\text{def}}{=}$ # of channels chosen from previously jammed channel space for jamming.

We have a constraint $\alpha_{C1}^t + \alpha_{D1}^t < N_1^t$ where $N_1^t = N - (I_C^t + I_D^t + J_C^t + J_D^t)$ gives the total number of uncollided and unjammed channels. We also have $\alpha_{C2}^t + \alpha_{D2}^t < N_2^t$ where $N_2^t = I_C^t + I_D^t$ is the total number of collided channels, and $\alpha_{C3}^t + \alpha_{D3}^t < N_3^t$ where $N_3^t = J_C^t + J_D^t$ is the total number of jammed channels.

2) Combinatorial analysis: We should consider combinations of ($m_{C\{1,2,3\}}$, $m_{D\{1,2,3\}}$) and ($n_{C\{1,2,3\}}$, $n_{D\{1,2,3\}}$) subject to the constraints represented by I_C, I_D, J_C, and J_D. Using the binomial coefficient $\binom{n}{k} = \frac{n!}{k!(n-k)!}$, the probability of m_{C1} control and m_{D1} data channels collided given that BF and RF networks choose from *previously uncollided and unjammed* channels is:

$$p(m_{C1}, m_{D1} | I_C^t, I_D^t, J_C^t, J_D^t, a_B^t, a_R^t) = \frac{\binom{\alpha_{C1}^t}{m_{C1}} \binom{\alpha_{D1}^t}{m_{D1}} \binom{N_1^t - \alpha_{C1}^t - \alpha_{D1}^t}{\alpha_{I1}^t + \alpha_{J1}^t - m_{C1} - m_{D1}}}{\binom{N_1^t}{\alpha_{I1}^t + \alpha_{J1}^t}}$$

The probability of m_{C2} control and m_{D2} data channels collided given that BF and RF networks choose from *previously collided* channels is:

$$p(m_{C2}, m_{D2} | I_C^t, I_D^t, J_C^t, J_D^t, a_B^t, a_R^t) = \frac{\binom{\alpha_{C2}^t}{m_{C2}} \binom{\alpha_{D2}^t}{m_{D2}} \binom{N_2^t - \alpha_{C2}^t - \alpha_{D2}^t}{\alpha_{I2}^t - m_{C2} - m_{D2}}}{\binom{N_2^t}{\alpha_{I2}^t}}$$

The probability of m_{C3} control and m_{D3} data channels collided given that BF and RF networks choose from *previously jammed* channels is:

$$p(m_{C3}, m_{D3} | I_C^t, I_D^t, J_C^t, J_D^t, a_B^t, a_R^t) = \frac{\binom{\alpha_{C3}^t}{m_{C3}} \binom{\alpha_{D3}^t}{m_{D3}} \binom{N_3^t - \alpha_{C3}^t - \alpha_{D3}^t}{\alpha_{J2}^t - m_{C3} - m_{D3}}}{\binom{N_3^t}{\alpha_{J2}^t}}$$

The probability of n_{C1} control and n_{D1} data channels jammed given that BF and RF networks choose from *previously uncollided and unjammed* channels is:

$$p(n_{C1}, n_{D1} | I_C^t, I_D^t, J_C^t, J_D^t, a_B^t, a_R^t) = \frac{\binom{\alpha_{C1}^t}{n_{C1}} \binom{\alpha_{D1}^t}{n_{D1}} \binom{N_1^t - \alpha_{C1}^t - \alpha_{D1}^t}{\alpha_{I1}^t + \alpha_{J1}^t - n_{C1} - n_{D1}}}{\binom{N_1^t}{\alpha_{I1}^t + \alpha_{J1}^t}}$$

The probability of n_{C2} control and n_{D2} data channels jammed given that BF and RF networks choose from *previously collided* channels is:

$$p(n_{C2}, n_{D2} | I_C^t, I_D^t, J_C^t, J_D^t, a_B^t, a_R^t) = \frac{\binom{\alpha_{C2}^t}{n_{C2}} \binom{\alpha_{D2}^t}{n_{D2}} \binom{N_2^t - \alpha_{C2}^t - \alpha_{D2}^t}{\alpha_{I2}^t - n_{C2} - n_{D2}}}{\binom{N_2^t}{\alpha_{I2}^t}}$$

The probability of n_{C3} control and n_{D3} data channels jammed given that BF and RF networks choose from *previously jammed* channels is:

$$p(n_{C3}, n_{D3} | I_C^t, I_D^t, J_C^t, J_D^t, a_B^t, a_R^t) = \frac{\binom{\alpha_{C3}^t}{n_{C3}} \binom{\alpha_{D3}^t}{n_{D3}} \binom{N_3^t - \alpha_{C3}^t - \alpha_{D3}^t}{\alpha_{J2}^t - n_{C3} - n_{D3}}}{\binom{N_3^t}{\alpha_{J2}^t}}$$

3) Posterior distribution: The combinatorial analysis leads to the posterior state transition probability distribution for CMNG presented in Eq. (2). To solve for an optimal strategy, we need to evaluate this posterior distribution. Unfortunately, the dynamic settings of CMNG (*e.g.*, changes in number of channels, comm nodes, jammers) make the analytical computation difficult. Moreover, it would be impractical to rework Eq. (2) whenever a CMNG parameter changes or nodes join and exit their network. We can alternatively sample the distribution, using a statistically rigorous technique such as Markov Chain Monte Carlo (MCMC); however, the MCMC

performance relies on the choice of a proposal distribution that must work well for CMNG, which by itself is an active area of research. In the next section, we propose Q-learning [9] based methods that can avoid complex state transition computations by a technique called *value iteration* [10].

IV. Determining Optimal Strategies with Q-learning

As a decision maker, the agent in Q-learning has a choice to maximize the reward by choosing the best known action or trying out one of the other actions in the hope of better payoffs in the long run. The former strategy is termed *exploitation*, and the latter *exploration*. In this section, we propose three comparable methods based on Minimax-Q [11], Nash-Q [12], and Friend-or-foe Q [13] learning algorithms that can solve for optimal antijamming and jamming strategies in CMNG.

A. Q-learning Background

Q-learning evaluates the *quality* of an action possible at a particular state and the *value* of that state. The Bellman equations characterize such optimization:

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} p(s'|s,a) V(s') \tag{3}$$

$$V(s) = \max_{a'} Q(s,a') \tag{4}$$

The key strength of Q-learning is the value iteration technique that an agent performs an update $Q(s,a) = R(s,a) + \gamma V(s')$ in place of Eq. (3) without explicit knowledge of transition probability $p(s'|s,a)$. We remind that a strategy π is the probability distribution of actions a at state s. Linear programming can solve for $\pi^* = \arg\max_\pi \sum_a Q(s,a)\,\pi$ in place of Eq. (4).

B. Decomposition of CMNG

The coexistence of the two opposing kinds (*i.e.*, comm and jammer) in BF and RF networks decomposes CMNG into two subgames, namely *antijamming* and *jamming* games. Fig. 3 illustrates the antijamming-jamming relationship among the nodes. In antijamming game, the BF comm nodes strive to maximize their throughput primarily by avoiding hostile jamming from the RF jammers. Additionally, imperfect coordination within the BF network that causes a BF jammer to jam its own BF comm node (*i.e.*, misjamming) should be avoided. Collision avoidance among comm nodes is another objective of antijamming game.

In jamming game, the BF jammers try to minimize the RF data throughput by choosing the best channels to jam. A BF jammer can target a data channel frequently accessed by the RF comm nodes or alternatively aims for an RF control channel, which would result a small immediate reward but a potentially larger value in the future by blocking RF data traffic. Misjamming avoidance is also an objective for jamming game. For BF network, the primary means to avoid misjamming in jamming game is to coordinate the actions of the BF jammers. This is different for the case of antijamming game where the avoidance is done by coordinating the actions of the BF comm nodes.

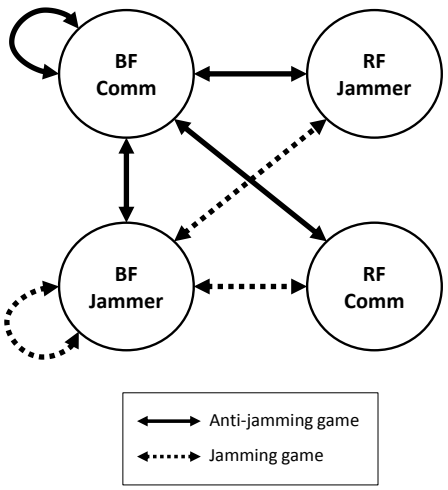

Fig. 3. Antijamming and jamming relationship

C. Minimax-Q Learning for CMNG

Minimax-Q assumes a zero-sum game that implies $Q_B(s^t, a_B^t, a_R^t) = -Q_R(s^t, a_B^t, a_R^t) = Q(s^t, a_B^t, a_R^t)$. This holds tightly for the CMNG jamming subgame where the jammer's gain is precisely the comm throughput loss of the opponent. In order to solve antijamming and jamming subgames jointly, we propose a slight modification to the original Minimax-Q algorithm in Littman [11]. First, we divide the strategy of BF network π_B into its antijamming and jamming substrategies, π_{B1} and π_{B2}. Then, we add an extra $\mathrm{minimax}$ operator to our value function in Eq. (5). The modified Q-function in Eq. (6) can be computed iteratively, using Eqs. (7) and (8). α^t gives the learning rate that decays over time, $\alpha^{t+1} = \alpha^t \cdot \delta$ for $0 < \delta < 1$.

D. Nash-Q Learning for CMNG

Nash-Q [12] can solve a general-sum game in addition to zero-sum games. This makes an important distinction to Minimax-Q although the Nash-Q value function for a zero-sum game in Eq. (9) is different from Eq. (5) by only one extra term $\hat{\pi}_R(a_R^t)$. This means that Nash-Q requires to estimate the policy of the opponent's agent. For CMNG, the BF agent needs to learn $\hat{\pi}_{R1}$ and $\hat{\pi}_{R2}$, the antijamming and jamming substrategies of RF network. The Q-function for the zero-sum Nash-Q is given by Eq. (10). For a general-sum game, the BF agent should compute Q_B and Q_R separately at the same time while observing its reward $r_B^t = r_B(s^t, a_B^t, a_R^t)$ and estimating the RF r_R^t by Eqs. (11) and (12). Nash-Q emphasizes the finding of a joint *equilibrium* under the mixed strategies $(\pi_B, \hat{\pi}_R)$.

E. Friend-or-foe Q-learning (FFQ) for CMNG

Although Nash-Q is applicable to both zero-sum and general-sum games, its convergence guarantee is considered too restrictive [13]. Game-theoretically, Friend-or-foe Q-learning (FFQ) introduced in Littman 2001 [13] does not solve any new problem. FFQ is a computational enhancement

$$
\begin{aligned}
&p(I_C^{t+1}, I_D^{t+1}, J_C^{t+1}, J_D^{t+1} | I_C^t, I_D^t, J_C^t, J_D^t, a_B^t, a_R^t) \\
&= \sum_{\substack{I_C^{t+1}=m_{C1}+m_{C2}+m_{C3} \\ I_D^{t+1}=m_{D1}+m_{D2}+m_{D3} \\ J_C^{t+1}=n_{C1}+n_{C2}+n_{C3} \\ J_D^{t+1}=n_{D1}+n_{D2}+n_{D3}}} p(m_{C1}, m_{D1} | I_C^t, I_D^t, J_C^t, J_D^t, a_B^t, a_R^t) \times p(m_{C2}, m_{D2} | I_C^t, I_D^t, J_C^t, J_D^t, a_B^t, a_R^t) \\
&\qquad \times p(m_{C3}, m_{D3} | I_C^t, I_D^t, J_C^t, J_D^t, a_B^t, a_R^t) \times p(n_{C1}, n_{D1} | I_C^t, I_D^t, J_C^t, J_D^t, a_B^t, a_R^t) \\
&\qquad \times p(n_{C2}, n_{D2} | I_C^t, I_D^t, J_C^t, J_D^t, a_B^t, a_R^t) \times p(n_{C3}, n_{D3} | I_C^t, I_D^t, J_C^t, J_D^t, a_B^t, a_R^t) \\
&= \sum_{\substack{I_C^{t+1}=m_{C1}+m_{C2}+m_{C3} \\ I_D^{t+1}=m_{D1}+m_{D2}+m_{D3} \\ J_C^{t+1}=n_{C1}+n_{C2}+n_{C3} \\ J_D^{t+1}=n_{D1}+n_{D2}+n_{D3}}} \frac{\binom{\alpha_{C1}^t}{m_{C1}}\binom{\alpha_{D1}^t}{m_{D1}}\binom{N_1^t-\alpha_{C1}^t-\alpha_{D1}^t}{\alpha_{J1}^t+\alpha_{J1}^t-m_{C1}-m_{D1}}}{\binom{N_1^t}{\alpha_{I1}^t+\alpha_{J1}^t}} \times \frac{\binom{\alpha_{C2}^t}{m_{C2}}\binom{\alpha_{D2}^t}{m_{D2}}\binom{N_2^t-\alpha_{C2}^t-\alpha_{D2}^t}{\alpha_{I2}^t-m_{C2}-m_{D2}}}{\binom{N_2^t}{\alpha_{I2}^t}} \\
&\qquad \times \frac{\binom{\alpha_{C3}^t}{m_{C3}}\binom{\alpha_{D3}^t}{m_{D3}}\binom{N_3^t-\alpha_{C3}^t-\alpha_{D3}^t}{\alpha_{J2}^t-m_{C3}-m_{D3}}}{\binom{N_3^t}{\alpha_{J2}^t}} \times \frac{\binom{\alpha_{C1}^t}{n_{C1}}\binom{\alpha_{D1}^t}{n_{D1}}\binom{N_1^t-\alpha_{C1}^t-\alpha_{D1}^t}{\alpha_{I1}^t+\alpha_{J1}^t-n_{C1}-n_{D1}}}{\binom{N_1^t}{\alpha_{I1}^t+\alpha_{J1}^t}} \\
&\qquad \times \frac{\binom{\alpha_{C2}^t}{n_{C2}}\binom{\alpha_{D2}^t}{n_{D2}}\binom{N_2^t-\alpha_{C2}^t-\alpha_{D2}^t}{\alpha_{I2}^t-n_{C2}-n_{D2}}}{\binom{N_2^t}{\alpha_{I2}^t}} \times \frac{\binom{\alpha_{C3}^t}{n_{C3}}\binom{\alpha_{D3}^t}{n_{D3}}\binom{N_3^t-\alpha_{C3}^t-\alpha_{D3}^t}{\alpha_{J2}^t-n_{C3}-n_{D3}}}{\binom{N_3^t}{\alpha_{J2}^t}}
\end{aligned} \tag{2}
$$

$$
V(s^t) = \max_{\pi_{B1}(A_{B,comm})} \min_{a_{R,jam}^t} \max_{\pi_{B2}(A_{B,jam})} \min_{a_{R,comm}^t} \sum_{a_B^t} Q(s^t, a_B^t, a_R^t)\, \pi_B(a_B^t) \tag{5}
$$

$$
\begin{aligned}
Q(s^t, a_B^t, a_R^t) &= r(s^t, a_B^t, a_R^t) + \gamma \sum_{s^{t+1}} T(s^t, a_B^t, a_R^t, s^{t+1})\, V(s^{t+1}) \\
&= r(s^t, a_B^t, a_R^t) + \gamma \sum_{s^{t+1}} p(s^{t+1} | s^t, a_B^t, a_R^t)\, V(s^{t+1})
\end{aligned} \tag{6}
$$

$$
Q(s^t, a_B^t, a_R^t) = (1-\alpha^t) Q(s^t, a_B^t, a_R^t) + \alpha^t [r(s^t, a_B^t, a_R^t) + \gamma V(s^{t+1})] \tag{7}
$$

$$
\begin{aligned}
Q(s^t, a_B^t, a_R^t) &= (1-\alpha^t)\, Q(s^t, a_B^t, a_R^t) \\
&+ \alpha^t [r(s^t, a_B^t, a_R^t) + \gamma \max_{\pi_{B1}(A_{B,comm})} \min_{a_{R,jam}^t} \max_{\pi_{B2}(A_{B,jam})} \min_{a_{R,comm}^t} Q(s^t, a_B^t, a_R^t)\, \pi_B(a_B^t)]
\end{aligned} \tag{8}
$$

$$
V(s^t) = \max_{\pi_{B1}(A_{B,comm})} \min_{\hat{\pi}_{R2}(A_{R,jam})} \max_{\pi_{B2}(A_{B,jam})} \min_{\hat{\pi}_{R1}(A_{R,comm})} \sum_{a_B^t} Q(s^t, a_B^t, a_R^t)\, \pi_B(a_B^t)\, \hat{\pi}_R(a_R^t), \tag{9}
$$

$$
\begin{aligned}
Q(s^t, a_B^t, a_R^t) = (1-\alpha^t)\, Q(s^t, a_B^t, a_R^t) + \ \alpha^t [r(s^t, a_B^t, a_R^t) \ + \\
\gamma \max_{\pi_{B1}(A_{B,comm})} \min_{\hat{\pi}_{R2}(A_{R,jam})} \max_{\pi_{B2}(A_{B,jam})} \min_{\hat{\pi}_{R1}(A_{R,comm})} Q(s^t, a_B^t, a_R^t)\, \pi_B(a_B^t)\, \hat{\pi}_R(a_R^t)]
\end{aligned} \tag{10}
$$

$$
\begin{aligned}
Q_B(s^t, a_B^t, a_R^t) = (1-\alpha^t)\, Q_B(s^t, a_B^t, a_R^t) + \ \alpha^t [r(s^t, a_B^t, a_R^t) \ + \\
\gamma \max_{\pi_{B1}(A_{B,comm})} \min_{\hat{\pi}_{R2}(A_{R,jam})} \max_{\pi_{B2}(A_{B,jam})} \min_{\hat{\pi}_{R1}(A_{R,comm})} Q_B(s^t, a_B^t, a_R^t)\, \pi_B(a_B^t)\, \hat{\pi}_R(a_R^t)]
\end{aligned} \tag{11}
$$

$$
\begin{aligned}
Q_R(s^t, a_B^t, a_R^t) = (1-\alpha^t)\, Q_R(s^t, a_B^t, a_R^t) + \ \alpha^t [r(s^t, a_B^t, a_R^t) \ + \\
\gamma \max_{\pi_{B1}(A_{B,comm})} \min_{\hat{\pi}_{R2}(A_{R,jam})} \max_{\pi_{B2}(A_{B,jam})} \min_{\hat{\pi}_{R1}(A_{R,comm})} Q_R(s^t, a_B^t, a_R^t)\, \pi_B(a_B^t)\, \hat{\pi}_R(a_R^t)]
\end{aligned} \tag{12}
$$

---

and provides better convergence properties by relaxing the restrictive conditions of Nash-Q. For this relaxation, FFQ requires extra information that other agents in the game should be classified *friendly cooperative* or *hostile*.

TABLE IV
SUMMARY OF SIMULATION SETUP

Parameter	Description	Value used
N	# of channels	10
N_{comm}	# of comm nodes per network	2
N_{jam}	# of jammers per network	2
p_{Tx}	Node's Tx probability	1
B	Reward for successful Tx	1
τ	Total # of time slots simulated	2,000

In FFQ, the BF agent maintains only one Q-function:

$$Q_B(s^t, a_B^t, a_R^t) = (1-\alpha^t)\, Q_B(s^t, a_B^t, a_R^t) + \alpha^t [r(s^t, a_B^t, a_R^t) + \gamma \Psi_B] \quad (13)$$

If the opponent (RF agent) is identified as a friend, the Q-function for the BF network is updated by

$$\Psi_B = \max_{a_B^t, a_R^t} Q_B(s^t, a_B^t, a_R^t) \quad (14)$$

On the other hand, if the opponent is considered a foe, the Q-function is updated under the minimax criterion

$$\Psi_B = \max_{\pi_B(A_B)} \min_{\hat{\pi}_R(A_R)} \sum_{a_B^t} Q_B(s^t, a_B^t, a_R^t)\, \pi_B(a_R^t) \quad (15)$$

V. EVALUATION

In this section, we evaluate the performance of Minimax-Q, Nash-Q, and FFQ learning based strategies under the 2-network CMNG of Blue and Red Forces.

A. Implementation

We have implemented Minimax-Q, Nash-Q, and FFQ learning algorithms in MATLAB, using `linprog` function from Optimization Toolbox. We require to maintain the Q table, which is a three-dimensional array that can be looked up using state, BF and RF action vectors. At the end of each time slot, we compute the next state from the sensing result of each channel. Recall that state computation is done by counting I_C, I_D, J_C, and J_D parameters described in Section II.B. The action vector space is discrete, and we have pre-generated and indexed all possible action vectors for BF and RF. A strategy π is a two-dimensional array indexed by state and action vector (either BF or RF). The V table for the value function is indexed only by state.

The key is to integrate the updates for Q and V tables with a linear program that finds the optimal distribution π at each iteration. The procedure (for BF) is summarized below.

1) At current state `s`, choose `a_BF` according to `pi[s,:]` and execute
2) Sense RF node actions `a_RF`, observe instantaneous reward `r`, and compute next state `s'`
3) Update `Q[s,a_BF,a_RF]`
4) Solve linear program to rebalance `pi[s,:]`
5) Update `V[s]`
6) Decay learning rate `alpha`, transit to `s'` by `s=s'`, and go back to Step 1 and repeat

We rewrite the minimax optimization

$$\max \left[\min \sum_i Q_i \pi_i \right] \text{ s.t. ...}$$

to be solved by `linprog` to:

$$\max\, y \text{ s.t. } y \le \sum_i Q_i \pi_i, \text{ ...}$$

Modified V-functions in Eqns. (5) and (9) feature double $\max\min$ operators due to splitting CMNG into two subgames. There are two ways to solve these double $\max\min$ optimizations. First, we can assign a priority for each subgame and solve the higher priority subgame first (*e.g.*, relax π_{B1} before π_{B2}). This approach, however, requires to solve two linear programs in series. We can instead bind $a_{B,comm}$ and $a_{B,jam}$ into one vector after disallowing some obviously harmful actions between a comm node and jammer in the same team (*e.g.*, actions lead to misjamming) and solve only one linear program per iteration. Our results are based on the second approach.

B. Simulation Setup

Table IV describes simulation parameters and the values used. The spectrum under competition has $N = 10$ channels. Both BF and RF networks have 2 comm nodes and 2 jammers. We set each node's Tx probability $p_{\text{Tx}} = 1$. Therefore, all nodes in BF and RF networks transmit at every time slot. Upon a successful (*i.e.*, uncollided and unjammed) transmission, the comm node earns a reward B for its network. Similarly, the network for a jammer receives B when the jammer makes a successful jamming. We normalize B to 1, which translates to the maximum possible reward of 4 for each network at each time slot. For example, when all two BF jammers successfully jam the two RF comm nodes and the BF comm nodes transmit without collision or being jammed by RF jammers, BF network will receive a reward value of 4. Each network is assumed to use only one control channel. When Q-learning is used and the control channel gets jammed, the agent will receive no information update, halt in the next time slot, and not compute V- and Q-functions or π. We simulate each run for 2,000 time slots and observe reward performances.

C. Experimental Scenarios

We configure BF network to run strategies based on Q-learning and RF network to run simple, non-learning strategies *static* and *random*. Under the static strategy, RF comm nodes and jammers act on statically pre-configured channels that remain the same during a simulation. Under the random algorithm, the RF nodes choose uniformly random channels at each time slot. We have simulated all 6 possible scenarios for CMNG between BF and RF networks:

1) Minimax-Q (BF) vs. Static (RF)
2) Nash-Q (BF) vs. Static (RF)
3) FFQ (BF) vs. Static (RF)
4) Minimax-Q (BF) vs. Random (RF)
5) Nash-Q (BF) vs. Random (RF)
6) FFQ (BF) vs. Random (RF)

D. Results and Discussion

We adopt average cumulative reward of a network over time as the performance evaluation metric:

$$\bar{R}_\tau = \frac{1}{\tau}\sum_{t=1}^{\tau}\sum_{k=1}^{N_{tot}} r_k^t \tag{16}$$

where τ is the count of simulated time slots, and r_k^t the reward from kth node in the network at time t. Note the total number of nodes per network $N_{tot} = N_{comm} + N_{jam}$. Hence, the metric $\bar{R}_\tau$ reflects both the comm and jammer rewards. In Fig. 4, we plot the cumulative average rewards for BF network operating Q-learning based methods Minimax-Q, Nash-Q, and FFQ against RF network's static strategy over time. Fig. 5 depicts the cumulative average rewards for BF network operating under Minimax-Q, Nash-Q, and FFQ based strategies against RF network's random strategy.

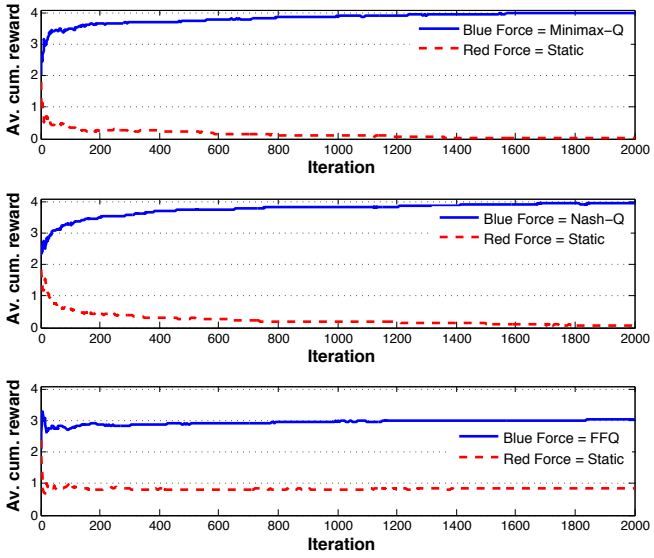

Fig. 4. Q-learning vs. Static

Under the simulation parameters that we have chosen, the Q-learning algorithms converge to a steady-state distribution of the BF actions within 1,000 iterations. Under such convergence, the BF average cumulative reward metric seems to approach to an asymptotically optimal value. We observe that the minimax criterion results in a more aggressive strategy than Nash-Q: 1) Minimax-Q converges to a steady-state cumulative average reward value faster; and 2) it outperforms Nash-Q by achieving slightly higher rewards over time. Static strategy has almost no chance against the learning algorithms as its steady-state average cumulative reward approaches to zero. On the contrary, learning seems harder against the random strategy particularly due to its effectiveness in jamming.

When running Minimax-Q or Nash-Q, we have configured the BF network with the centralized control, having a single agent that strategizes for the whole network. This means that the agent makes all access and jamming decisions in the network under an assumption that the nodes collaboratively sense

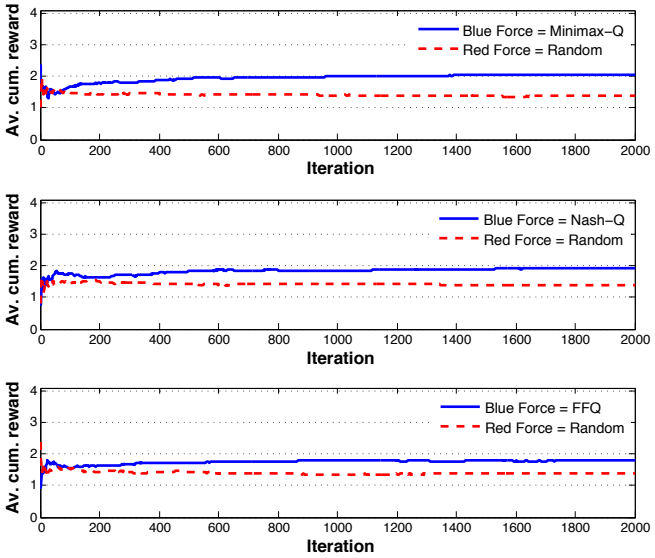

Fig. 5. Q-learning vs. Random

channels and observe the outcome, the agent can collect this information to facilitate Q-learning, and the nodes cooperate by following the agent's decision.

For FFQ, we have configured each BF node to be an agent. This represents a scenario with the distributed control where each node computes its own strategy. It is important to understand that FFQ becomes identical to Minimax-Q under our centralized control model because there are no other friend agents to the sole agent under the centralized control strategizing for the entire BF network, thus FFQ resorts to using only the Foe-Q function in Eq. (15), which is the same as the Minimax-Q function. Therefore, the use of FFQ learning in CMNG makes sense for distributed control scenarios only.

There is no explicit cooperation among the nodes in the distributed scenario for FFQ, and we have only provided each BF node with information whether some node it senses on a channel is a friend (*i.e.*, another BF node) or foe (*i.e.*, an RF comm node or jammer). Interestingly, with such knowledge, FFQ (despite under the distributed control) can achieve a good performance that is comparable to Minimax-Q or Nash-Q in the centralized setting where the information collected by each node is conveniently made available to the network's singular policy maker. This suggests that FFQ is the most viable choice for a network that lacks the centralized control (*e.g.*, MANET) among the three Q-learning techniques considered.

VI. Related Work

Reinforcement learning [14] extends beyond the postulate of Markov Decision Process that an agent's environment is stationary and contains no other agents. The original concept of Q-learning was introduced by Watkins and Dayan [9]. Littman [11] proposed Minimax-Q learning for a zero-sum two-player game. Littman and Szepesvári [10] showed that Minimax-Q converges to the optimal value suggested by game theory. Hu and Wellman [12] described Nash-Q that was

distinguished from Minimax-Q by solving a general-sum game with a Nash equilibrium computation in its learning algorithm. Nash-Q has more general applicability, but its assumptions on the sufficient conditions for convergence guarantee are known to be restrictive. Friend-or-foe Q-learning (FFQ) [13] converges precisely to the steady-state value that Nash-Q guarantees. The key improvement of FFQ is relaxation of the restrictive conditions that Nash-Q has, but FFQ requires *a priori* knowledge on other agents identified as either a friend or foe.

This paper considers some similar problems discussed by Wang *et al.* [2] such as finding a strategy against hostile jamming. They 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. Our work is novel and differentiated from existing work by the following. We have brought in friendly jammers to provide an integrated, stochastic antijamming-jamming game played between two competing cognitive radio networks. We embrace jamming as a means to compete in a hostile environment typically assumed in tactical mobile networking. At the same time, we try to best the enemy jammers that pose a serious threat to the ally comm activities. We promote the notion of strategic jamming enabled by reinforcement learning. We modify existing Q-learning algorithms to solve for optimal antijamming and jamming strategies jointly.

VII. Conclusion

We have seen promising applications of cognitive radio in commercial domains that suggest new, more intelligent approaches to utilize spectrum resource. There is a growing interest to leverage agile capabilities of cognitive radio for tactical networking, and in this paper we have investigated the competition and coexistence among cognitive radio nodes that form networks in an attempt to maximize their objective. We have considered two different types of radio devices, namely comm node and jammer, and studied the interaction of their common and conflicting interests in a stochastic game framework. In particular, we have applied reinforcement Q-learning techniques to strategize optimal channel accessing schemes for comm nodes and jammers to cope with a hostile environment possessing the same capabilities. Our results indicate that Minimax-Q learning is more suitable for an aggressive environment than Nash-Q. More interestingly, Friend-or-foe Q-learning is most feasible for distributed mobile ad hoc networking scenarios that can hardly expect centralized control.

We plan to build a prototype system that can be deployed in the CMNG environment ultimately. Our immediate future work includes algorithmic improvements to scale the number of nodes in a network efficiently, adding more friendly and enemy networks to the current two-network model, rigorous analysis on the accidental use of incorrect information (due to sensing imperfections) in learning, and design of system components such as cognitive sensing and jamming detection at the physical and MAC layers. We also envision to enhance our computational framework through more robust linear programming methodologies and parallelization.

Acknowledgment

We thank Jason Hiebel of Michigan Technological University for earlier foundation of this work during his research internship at MIT Lincoln Laboratory in the summer of 2012.

References

[1] Q. Zhao and B. Sadler, "A Survey of Dynamic Spectrum Access," *IEEE Signal Processing Magazine*, May 2007.

[2] 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.

[3] H. Li and Z. Han, "Dogfight in Spectrum: Combating Primary User Emulation Attacks in Cognitive Radio Systems, Part I," *IEEE Trans. on Wireless Communications*, vol. 9, no. 11, pp. 3566–3577, 2010.

[4] S. Sodagari and T. Clancy, "An Anti-jamming Strategy for Channel Access in Cognitive Radio Networks," in *Decision and Game Theory for Security*. Springer LNCS, 2011, vol. 7037, pp. 34–43.

[5] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks," in *Proc. of ACM MobiHoc*, 2005.

[6] M. Pajic and R. Mangharam, "Anti-jamming for Embedded Wireless Networks," in *Proc. of IPSN*, 2009.

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

[8] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

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

[10] M. L. Littman and C. Szepesvári, "A Generalized Reinforcement-learning Model: Convergence and Applications," in *Proc. of International Conference on Machine Learning (ICML)*, 1996.

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

[12] 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.

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

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