# Regret Minimization-based Robust Game Theoretic Solution for Dynamic Spectrum Access

Yalin Sagduyu*, Yi Shi*, Allen B. MacKenzie‡ and Y. Thomas Hou‡
* Intelligent Automation, Inc., Rockville, MD, USA
‡ Virginia Tech, Blacksburg, VA, USA
Email: {ysagduyu, yshi}@i-a-i.com, {mackenab, thou}@vt.edu

***Abstract*—This paper presents a game theoretic solution for hierarchical spectrum sharing between primary users (PUs) and secondary users (SUs) in the presence of cognitive interferers. There exist several forms of uncertainty, including channel conditions, packet traffic, and spectrum occupancy of users across channels. This uncertainty is further aggravated by delays (such as propagation delays observed over satellite links) that make spectrum-efficient communication a challenging task. A robust game theoretic framework is developed for dynamic spectrum access (DSA) management over multiple channels. Cognitive functionalities employed in the game solution include selecting channels for data transmission and performing power control at each user to sustain target rates. By considering random utility functions, the game engine based on regret minimization provides low complexity and fast solutions compared to traditional game solutions based on expected utility maximization. Detailed numerical results with comparison to benchmark schemes (the ideal case and the random case where users have perfect or no knowledge on channel availability, respectively) are provided to show the effectiveness of robust game theory-enabled approach.**

***Keywords*—Dynamic spectrum access; hierarchical spectrum sharing; cognitive radio; game theory; regret minimization; uncertainty.**

## I. Introduction

We consider hierarchical spectrum sharing between primary users (PUs) with dedicated channels and secondary users (SUs) opportunistically accessing available channels in the presence of cognitive interferers. The focus is on uplink transmissions from PUs or SUs to a receiver which may represent a satellite in a satellite communication (SATCOM) system. Each PU has assigned channels for its own traffic with random packet arrivals. Each SU can sense and access idle channels not used by PUs while each cognitive interferer can sense PU and SU transmissions and interfere with detected transmissions.

By taking into account delays in acknowledgments following data transmissions (such as those caused by the propagation delays over satellite links), we design the game engine for each type of user (PUs, SUs and cognitive interferers). We apply *regret minimization* [1] as a low-complexity and fast alternative to *expected utility maximization* [2] in the form of Bayesian games. We define a policy space and utility for each type of users. The defined utility is a random function, since both reward for successful transmissions and cost due to delay, channel activation and channel reservation are random variables. We apply regret minimization techniques to choose a policy randomly based on its regret, which is the additional cost due to not using the optimal policy. We present a decision making module for each type of user, i.e. PU, SU, and cognitive interferer, each with its own policy space and utility. The strategies of users include selecting channels and performing power control.

1) A PU has data packets that arrive at random intervals to be transmitted to the receiver by using its assigned channels. A PU performs handshaking with the receiver before transmission. Due to round trip delay, a channel cannot be available immediately after the PU sends a Request to Send (RTS). Thus, channels are classified as idle (to be activated), activation pending, and available. Once a channel is available, the PU can transmit on the channel, otherwise the channel will become idle again. PUs need to decide which idle channels should be activated, which available channels should be used for transmissions, and which available channels should be reserved for future transmissions. We define a policy space to make these decisions. We also define a utility function for PUs, which considers both reward for successful transmissions and cost for delay, activation, and reservation.
2) An SU always has packets to be transmitted on channels that are sensed as idle. It also performs handshaking with the receiver before transmission. Note that if requests from both the PU and an SU are received, the receiver will send a Clear to Send (CTS) to the PU. SUs need to decide which idle channels should be activated. We define a policy space to make this decision. We also define a utility function for SUs, which considers both reward for successful transmissions and cost for delay and activation.
3) An interferer can sense busy channels. Since we assume limited interferer capability, an interferer needs to select a subset of busy channels. We define a policy space to make this decision. The utility of an interferer is a function of interfered packets.

We then apply regret minimization for each user to select a suitable policy. This approach has low complexity and fast convergence speed. Effects of spectrum sensing on throughput and delay are studied. Results are also compared to an ideal (genie-based) policy and a random policy in which users have perfect or no knowledge on channel availability, respectively.

DISTRIBUTION A. Approved for public release; distribution unlimited. 377ABW-2014-1011. Supported by USAF/AFRL under contract FA9453-13-M-0155. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Air Force.

Regret minimization has been applied to cognitive networks at different levels [3]–[8]. The focus of [3] was on coexistence between macro- and femto- cell tiers, whereas we consider DSA with PUs, SUs, and interferers. [4] solved a one-shot scheduling problem of enabling the maximum number of concurrent transmissions. Instead, we address a general scheduling problem for DSA where different transmissions can be scheduled in different time slots. [5]–[7] considered PU and SU transmissions only. We also consider the impact of interferers. [8] studied a network of PUs, SUs and interferers but focused on the sensing problem. We consider both sensing and throughput maximization in this paper.

The rest of the paper is organized as follows. Section II provides the system model, including the network setting, and the PU, SU, and cognitive interferer models. Then Section III presents the game engine based on regret minimization for each type of node. Section IV provides the performance evaluation of the developed game engine compared to different alternatives. Finally, Section V concludes the paper.

## II. System Model

### A. Network Model

We consider an uplink network scenario. There are three types of nodes: PUs, SUs, and cognitive interferers:

1) PUs and SUs are transmitting to a receiver (e.g., a satellite). PUs have dedicated channels for data transmission while SUs can access channels only when PUs are not transmitting.
2) Cognitive interferers aim to interfere with the uplink PU/SU transmissions at the receiver.

### B. Secondary User (SU) Model

Each SU looks for opportunistic access to PU channels and aims to regulate its admission of new packets to a rate that can actually be supported. The opportunistic access requires that an SU senses some channels, finds idle channels (not in use by PUs), and then uses these idle channels for transmission.

An SU can sense a PU or another SU transmission by observing ACK/NAK. When a control packet is sensed on one channel, this channel is busy unless it was used by a PU and this PU has an empty queue. Thus, an SU may simply assume ACK/NAK indicates a busy channel.

In addition, an SU can sense PU/SU transmission either directly or by sensing sidelobes. For example, a PU/SU ground transmitter in SATCOM uses a directional antenna to communicate with the receiver. Although the majority of the power is transmitted in the main lobe towards the receiver, there is some power in sidelobes. An SU in a sidelobe can detect channel busy/idle status. Such information may not always be available, e.g., an SU is not in a sidelobe, and may have error since power in sidelobe is much less than interference. Thus, an SU uses such information as an auxiliary approach to improve the results by sensing ACK/NAK. Any spectrum sensing method such as energy detector can be used.

Once some channels are identified as available, an SU performs handshaking before its transmissions. Since there are various uncertainties in the system, e.g., uncertainty due to delayed channel status, the actual channel condition is unknown. The receiver will send an ACK/NAK to confirm the received data, along with channel status. Each ACK/NAK is delayed by RTT.

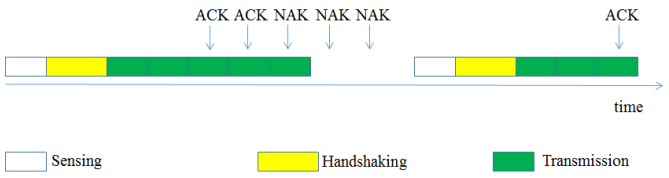


Fig. 1. Sample SU activities.

There are two cases of NAK for SU transmissions. The first case is the same as the NAK for PU transmissions i.e., a NAK is generated when the received SINR is less than a threshold $SINR_{\min}$. This case may indicate an interfered channel. The second case is that the receiver has received a PU handshaking request and thus terminates SU transmissions.

Fig. 1 shows a sample of SU activities on one channel. At the beginning, SU performs sensing, finds an available channel, performs handshaking and then transmits its data. Since an SU always has data to transmit, it transmits the second packet right after it finishes the transmission of the first packet. Then if an ACK is received, the SU knows that the current channel is still available and updates channel status. Otherwise, the SU knows that the current channel has become unavailable.

In addition to design possibility in this example, we consider more challenging cases in algorithm design, including 1) SU can sense multiple channels and transmit on multiple available channels, 2) sensing/handshaking/NAK may indicate that some channels are unavailable and thus not all sensed channels are available.

### C. Primary User (PU) Model

PU model is similar to SU model with the following differences: 1) Each PU has random packet arrivals. 2) A PU has a number of assigned channels and thus there is no sensing before handshaking.

### D. Cognitive Interferer Model

Interferer model is also similar to SU model with the difference that an interferer does not perform handshaking. It just interferes selected channels after sensing.

## III. Game Engine based on Regret Minimization

We developed different policies for PUs, SUs and interferers under the regret minimization framework [3]–[8], where both reward (function of random number of transmitted packets) and cost (function of random policy) are random functions. These policies are implemented as regret minimization strategies. In general, regret minimization strategies are known to converge to approximate correlated equilibria [1]. In some cases, these equilibria have significantly better social welfare properties than conventional Nash equilibria, which maximize expected utility. The full game model for the system that we have described is a stochastic game with partial observability.

Namely, the system evolves stochastically and the players cannot observe the entire state, but only various signals associated with the state. Such a game is the multi-player analog of a Partially Observable Markov Decision Process (POMDP). Therefore, expected utility maximization would have very high complexity and a tractable solution to compute an equilibrium is unknown. Instead, regret minimization provides a fast and low-complexity solution of random policy updates (without solving an optimization problem explicitly).

### A. System Model for Game Setting

We consider a time-slotted system consisting of $n$ channels that are allocated to each PU. There are $n_{PU}$ PUs and thus a total of $n \times n_{PU}$ channels. We consider $n_{SU}$ SUs and $n_J$ interferers. Each SU can sense and use any idle channels (not used by the corresponding PU and other SUs) to transmit data while each interferer can sense and interfere with any busy channels (used by PU/SU). For simplicity, we assume that a packet can be transmitted in exactly one slot, although the number of packets transmitted in one slot may be variable. The round-trip time between a ground station and the receiver is divided into $r$ slots. That is, data transmissions in slot $t$ are acknowledged in slot $t+r$. Similarly, we assume that it takes $r$ slots to activate an idle channel. When a user wants to activate a channel, it sends an activation request to the receiver in slot $t$, receives a response in slot $t+r$ and can begin transmitting on the channel in slot $t+r+1$.

Packet failures may still occur even when there is no interference, but the cause of these failures is unspecified (and therefore not modeled). In this case, in every slot a user (PU or SU) decides which subset of channels to use. Data can be sent on active channels immediately. As noted above, inactive channels that a user decides to activate in slot $t$ can be used for data beginning in slot $t+r+1$.

### B. PU's Strategy

The complete *state* for the PU at time $t$ consists of a state for each channel and the state of each packet in the PU's queue. The channel may be active (in which case the PU can send data on it in slot $t$), idle (in which case the PU can request to activate it), or in activation stage $k$, where $k \in \{1, 2, \ldots, r\}$.

Denote the number of channels in each state as $(n_0, n_1, \ldots, n_r, n_{r+1})$, where $n_0$ is the number of idle channels, $n_k$ is the number of channels in activation stage $k \in \{1, 2, \ldots, r\}$ and $n_{r+1}$ is the number of active channels. A complete state for each PU packet would indicate the "waiting stage" of any unacknowledged transmissions of the packet since it takes $r$ slots to learn the outcome of any transmission attempt. Denote the number of packets in each state as $(q_0, q_1, \ldots, q_r)$, where $q_0$ is the number of packets waiting for transmission and $q_k$ is the number of packets in waiting stage $k \in \{1, 2, \ldots, r\}$. A packet in waiting stage $k$ at time $t$ was transmitted in slot $t-k$ and an ACK (or NAK) is expected in slot $t+r-k$.

The set of *actions* available to the PU models deciding which of the idle channels to activate, which of the active channels (if any) to deactivate (by not using) and what packets to transmit on each of the active channels. The feedback that the PU receives in slot $t$ is the identity of any packets that succeeded, of those that were transmitted in slot $t-r$. Denote $a_{act}$ as the number of channels to activate and $a_{xmit}$ as the number of channels to transmit on, where

$$a_{act} \in \{0, 1, \ldots, n_0\} \text{ and } a_{xmit} \in \{0, 1, \ldots, n_{r+1}\}. \quad (1)$$

Moreover, we have

$$a_{act} > 0 \text{ only if } a_{xmit} = n_{r+1}. \quad (2)$$

When $q_0 \geq a_{xmit}$, $a_{xmit}$ packets in state 0 are transmitted. Otherwise, reservation packets are sent on $a_{xmit} - q_0$ channels for the sole purpose of keeping those channels active. The feedback $f^t$ received by the PU in slot $t$ is the number of packets that succeeded in slot $t-r$ and must satisfy

$$f^t \in \{0, \ldots, q_r^t = a_{xmit}^{t-r}\}. \quad (3)$$

The evolution for the channel states is specified by

$$n_k^{t+1} = \begin{cases} n_0^t - a_{act}^t + n_{r+1}^t - a_{xmit}^t, & k = 0, \\ a_{act}^t, & k = 1, \\ n_{k-1}^t, & k \in \{2, 3, \ldots, r\}, \\ a_{xmit}^t + n_r^t, & k = r+1. \end{cases} \quad (4)$$

For the packet queue, the evolution is specified by

$$q_k^{t+1} = \begin{cases} (q_0^t - a_{xmit}^t)^+ + q_r^t - f^t + a_0^t, & k = 0, \\ \min\{a_{xmit}^t, q_0^t\}, & k = 1, \\ q_{k-1}^t, & k \in \{2, 3, \ldots, r\}. \end{cases} \quad (5)$$

Here, $a_0^t$ is the number of new packet arrivals during slot $t$ and $x^+ = \max\{0, x\}$.

In each time slot, we assume that the PU receives a *reward* of one unit for each packet successfully delivered. The PU pays a delay penalty of $c_{delay}$ for each non-transmitted packet in queue $q_0$, where $0 < c_{delay} < 1$. Finally, the PU pays a cost $c_{rev}$ for each reservation packet transmitted, where $c_{rev} > 0$. We assume a channel activation cost $c_{act} > 0$. The introduction of $c_{act}$ is to prevent keeping all channels active, which will negatively impact SUs.

We assume that the PU wishes to maximize the average slot utility over time. In some sense, it does not matter which slot particular rewards and costs are credited to. But, when it comes to machine learning and regret minimization algorithms, this will make a significant difference. If the number of packets successfully acknowledged in time slot $t$, $f^t$ is credited as the reward in that time slot, then we end up with the awkward situation in which the reward in that slot is unconnected to the action taken in the slot, because it only depends on actions taken in slot $t-r$. So, instead, the reward in time slot $t$ is taken to be the number of packets that are successfully transmitted in slot $t$, that is $f^{t+r}$. This means that the reward in slot t is not known until slot $t+r$, but this is easier to handle in the learning framework than rewards that are decoupled from actions.

Thus, in time slot $t$, the PU *utility* is

$$\begin{aligned} u_i^t &= f^{t+r} - c_{delay}(q_0^t - a_{xmit}^t)^+ \\ &\quad - c_{rev}(a_{xmit}^t - q_0^t)^+ - c_{act} a_{act}^t. \end{aligned} \quad (6)$$

The PU's decision includes $a^t_{xmit}$ and $a^t_{act}$. The parameters include $r$, $n$, $c_{delay}$, $c_{rev}$, $c_{act}$, $a^t_0$, $q^0_k$ (usually assumed as zero), $n^0_k$ (usually assumed $n^0_0 = n$ and $n^0_k = 0$ for $k = 1, \ldots, r+1$) and $f^{t+r}$ (as a function of $a^t_{xmit}$). The PU strategy consists of two rules, an activation rule and a deactivation rule.

- *Activation rule*: Based on the current queue state and channel state, compute the number of channels that could definitely be used in slot $t+r+1$ if they were activated now (i.e., assuming no additional arrivals or failed packets). Call this number $n^t_{need}$. In slot $t$, activate enough channels to ensure that there will be at least $n_{need} + l_{act}$ active channels in slot $t+r+1$ subject to the number of available channels. Call the parameter $l_{act}$ the *activation level* of the algorithm.
- *Deactivation rule*: Idle channels may be deactivated provided that this will not cause the number of active channels to drop below $l_{maint}$ (the *maintenance level* of the algorithm).

Note that both parameters $l_{act}$ and $l_{maint}$ take values from the set $\{0, 1, \ldots, n\}$. Also note that $l_{maint} = n$ would be the strategy that keeps all channels active at all times. Furthermore, note that $l_{act} = l_{maint} = 0$ would be the conservative algorithm. An algorithm with positive values of the two parameters would behave in a way that seems intuitively sensible: It would activate some more channels than were currently "needed", in anticipation of new packet arrivals and failures, and it might keep some channels active if they were expected to be needed.

The two parameters for this algorithm can be easily learned via regret minimization. Namely, the PU can retrospectively evaluate the performance of the genie-aided algorithm and compare that to the performance of the developed algorithm with every possible parameter setting. This evaluation can be updated after each slot (once the outcomes for that slot are known $r$ slots later).

However, the regret minimization approach is generally not to simply choose the best parameter value based on past evaluation. Rather, the retrospective performance with different possible parameters can be used to choose a probability distribution over algorithm parameters, e.g., the polynomial weights algorithm [1]. There are standard ways to compute such a distribution. The idea is to weight the distribution towards the best performing values, but still have some diversity of action. The actual behavior chosen in each slot can then be determined by the algorithm using parameters drawn from the computed distribution. This stochastic selection has some advantages in a non-strategic environment, as it limits regret by broadening the action support, but it will have further advantages when play becomes strategic (i.e. when an interferer is introduced) as the randomization will make the PU's play less predictable by the interferer.

The set of all possible $(l_{act}, l_{maint})$ values is the PU *policy space*. We assume that the PU maintains a weight table of size $(n+1)\times(n+1)$, where each cell is associated with a particular choice of $l_{act}$ and $l_{maint}$. Initially, let this table be filled with values $w^0_{i,j} = 1$. When the outcomes for slot $t$ are known in slot $t+r$, the PU first evaluates what the genie-aided algorithm would have done in slot $t$ (i.e. how many channels it would have activated or deactivated), based on $q^t_k$,$n^t_k$,$a^{t+i}_0$ and what $q^{t+r+1}_0$ will be. Any deviation from this strategy will result in additional costs $c_{rev}$, $c_{delay}$ and $c_{act}$ beyond those charged to the genie-aided algorithm. For every cell $(i, j)$ in the weight table, the PU calculates the additional loss that would have resulted from using the algorithm with $l_{act} = i$ and $l_{maint} = j$ in that slot. Call this loss or *regret* value $l_{i,j}$. The weight table is then updated as

$$w^t_{i,j} = w^{t-1}_{i,j}(1 - \eta l_{i,j}), \tag{7}$$

where $\eta$ is the learning rate in the *polynomial weights algorithm* [1]. Typically, $\eta \ll 1/2$ and may itself be adaptive. It should be chosen to insure that the weights remain positive.

When deciding how many channels to activate or deactivate in slot $t+r+1$, the PU computes

$$W^t = \sum_{i=0}^{n}\sum_{j=0}^{n} w^t_{i,j} \tag{8}$$

and then selects the $(l_{act}, l_{maint})$ to use with a probability distribution with probability mass function

$$p^t_{i,j} = w^t_{i,j}/W^t. \tag{9}$$

In short, over time the PU learns the best values of the parameters to use to reduce its regret with respect to the genie-aided algorithm. Convergence of polynomial weights algorithm has been shown in [1]. By randomizing mostly over learned "good" parameter values, the expected regret is further reduced and an element of uncertainty is introduced. This uncertainty may have further benefits with respect to avoiding interferers. If this uncertainty turns out to be particularly desirable, then keeping a lower learning rate may be indicated.

So far, we discussed the decision process on channels only without consideration of power selection. After a PU selects channels, it will allocate its total transmit power on these channels such that the overall expected throughput can be maximized. In particular, its sensing module will determine the minimum equivalent isotropically radiated power (EIRP) on each channel to achieve target SINR, which in turn determines the minimum required power consumed on a channel. Then there are two cases. 1) The total required power on these selected channels is more than its total power, which means that some channels with large required power should not be used. 2) The total required power on these selected channels is less than its total power, which means that it can allocate more power on each channel. To improve SINR on all channels, it can allocate power proportional to minimum required power.

### C. SU's Strategy

SUs also apply regret minimization to develop a per-channel strategy. In particular, by observing past histories an SU can estimate the expected regret associated with each possible action (attempt to activate or not) for each channel that it does not currently possess in each time slot. Based on the estimated expected regret associated with each action, the SU will decide an activation probability for each channel. This process is similar to PU strategy. After an SU selects channels, it will allocate its total transmit power on these channels. This process is again similar to PU power allocation. We omit the discussion on detailed design of SU's strategy.

### D. Interferer Strategy

Assume that there are cognitive interferers that wish to interfere with the PU/SU's uplink transmissions. Further assume that an interferer is power limited and is only able to interfere with $j$ of the PU's $n$ uplink channels in each slot. If one assumes that there is still an independent and identically distributed (i.i.d.) failure probability of $p$ on the PU's remaining transmissions, then this means that an interferer can cap the total throughput of the PU at no more than $(n-j)(1-p)$ packets/slot. We consider the case that an interferer can receive ACK/NAK on all channels. Assume that an interferer's *utility* is based upon the number of packets successfully interfered. In our model, for slots more than $r$ slots old, an interferer knows whether or not it interfered and whether the outcome was ACK, NAK, or idle.

On the basis of these probability estimates, an interferer can choose the $j$ best channels to interfere with and allocate the same power on these channels. The parameter $m$ can be adjusted to optimize the tradeoff between the accuracy of the resulting estimates and the amount of time required to learn the estimates.

To compute regret, an interferer would regret interfering on any channel that wound up being idle and could evaluate various alternative strategies easily in retrospect. There is one aspect of the regret minimization strategy that might inspire interferers. Namely, always choosing the best $j$ channels to interfere with may be too predictable for PUs and SUs. Instead, an interferer should likely map its estimates of channel occupancy probability into a probability distribution over channels selected for interfering. In the game solution, this mapping would randomize interferer between channels offering "similar" likelihood of interfering success, adding some unpredictability to an interferer's behavior.

## IV. Performance Evaluation

### A. Setting for Performance Evaluation

For numerical results, we consider a general GEO receiver system at an altitude of 35,900 km operating at X-band or Ka-band. We assume channels each with 2.6 MHz bandwidth. Based on these properties, we can compute noise, path loss, antenna gain and general received power characteristics of the ground node-satellite communications [9].

*Noise.* The noise power $P_n$ is given by

$$P_n = kTW, \tag{10}$$

where $k$ is Boltzmann's constant ($k = 1.39 \times 10^{-23}$ J/K $= -228.6$ dBW/K/Hz), $T$ is the physical temperature of source in kelvin degrees, $W$ is the noise bandwidth in which the noise power is measured in Hz.

*Path loss.* The free space path loss is given by

$$L_p = 10 \log \left[ (4\pi R/\lambda)^2 \right], \tag{11}$$

where $R$ is the distance between the ground transmitter and the receiver and $\lambda$ is the wavelength (in meters) at the frequency of operation. We assume $R =$ 35,900km (by ignoring elevation angle effects) and wavelength $\lambda$ is approximately 0.0375m for X-band and 0.01m for Ka-band. Therefore, the free space path loss becomes $L_p = 201.6$dB for X-band and $L_p = 213$dB for Ka-band. In addition, there will be some atmospheric loss (e.g., 2 dB), clear air attenuation (e.g., 0.7dB), and rain attenuation (e.g., 6dB for 0.01% of the year). Suppose all these losses are summed to $L_a$, which needs to be learned in the process of channel estimation.

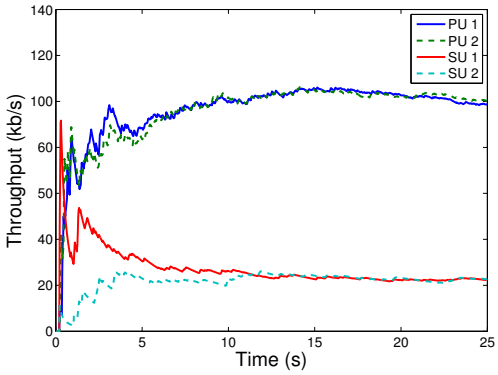


Fig. 2. Throughput performance of game solution.

*Antenna gain.* The antenna gain can be expressed as

$$G_t = 10 \log \left[ E_A (\pi D/\lambda)^2 \right], \tag{12}$$

where $D$ is the antenna diameter (assumed to be 2m for receiver and 5m for earth station), $\lambda$ is the wavelength (in meters) at the frequency of operation (0.0375m for X-band and 0.01m for Ka-band) and $E_A$ is the efficiency of antenna aperture (e.g., 68%). The transmit antenna gain is then computed as $G_t = 50.8$ dB for X-band and as $G_t = 62.3$ dB for Ka-band. Receiver antenna gain can be computed similarly. The receive antenna gain is then $G_r = 42.8$dB for X-band and as $G_r = 54.3$ dB for Ka-band. We assume 2m and 5m for antenna diameters of receiver and ground station, respectively, and 68% for the aperture efficiency.

*Received signal power.* The received signal power (in dB) is

$$P_r = P_t - L_p - L_a + G_t + G_r. \tag{13}$$

where transmit power at earth station is $P_t$. The total signal received at receive antenna is then $P_r + P_n$. In addition, there will be interference coming from other users (SUs and interferers), with total received interference power $P_I$.

### B. Performance Results

Suppose there are one receiver (satellite), two PUs, two SUs and two interferers (ground stations). Each PU has five assigned channels while SU/interferer can access all 10 channels. An interferer can interfere with at most three channels at the same time. The RTT, 0.239s, is divided into four slots. We assume that one channel can support transmission rate up to 100kb/s. The expected PU data arrival rate is 100kb/s.

In addition to our game solution, we also consider the following two cases. The *ideal case* means that each PU uses enough number of channels for its traffic, SUs share remaining channels, and each interferer can interfere with certain number of channels. The *random case* means that PUs and SUs have

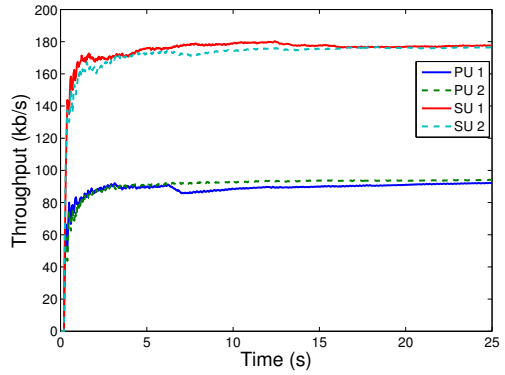


Fig. 3. Throughput performance under the ideal case.

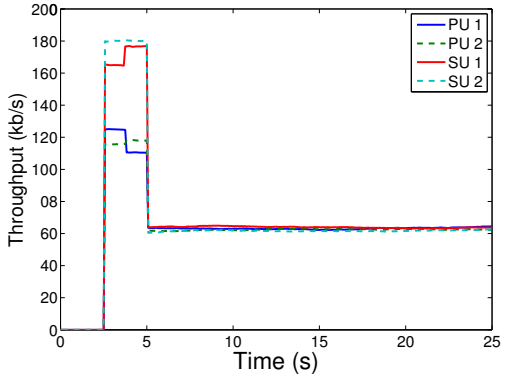


Fig. 4. Throughput performance under the random case.

no knowledge on channel availability and thus both PUs and SUs transmit on randomly selected channels.

Fig. 2, Fig. 3, and Table I show that a PU achieves the throughput value similar to its arrival rate under both the game solution and the ideal case. That is, PUs' performance is optimized under both solutions (achieving close values under numerical results shown in Table I). On the other hand, SUs' throughput under the game solution is much less than that under the ideal case, since the ideal case assumes no collision between SUs. This can be interpreted as a defense mechanism where the performance of SUs that face two levels of uncertainty (PU and jammer dynamics) is sacrificed to sustain the performance of PUs.

Fig. 2, Fig. 4, and Table I show that PUs achieve larger throughput values with the game solution than the random case while SUs achieve smaller throughput values with the game solution than the random case. This is mainly caused by random channel selection, i.e., SUs have more freedom to access channels but there is no protection for PU transmissions.

Table II shows similar delay performance for the game solution, the ideal case, and the random case. The ideal case has a slightly larger delay due to limited number of channels allocated to each user. Both PUs have a smaller delay with the game solution than the random case because SUs may interfere with PU transmissions in the random case. SUs have a much smaller delay with the game solution than the random case because SUs transmit more aggressively in the random case but many of these SU transmissions fail (and thus are retransmitted with large delay).

TABLE I. THROUGHPUT PERFORMANCE (IN KB/S) UNDER VARIOUS SCENARIOS.

| | PU1 | PU2 | SU1 | SU2 |
|---|---|---|---|---|
| Game solution | 98.7 | 100 | 23 | 22.6 |
| Ideal case | 92.2 | 94 | 177.6 | 176.6 |
| Random case | 64.4 | 62.3 | 63.8 | 62.5 |

TABLE II. DELAY PERFORMANCE (IN MS) UNDER VARIOUS SCENARIOS.

| | PU1 | PU2 | SU1 | SU2 |
|---|---|---|---|---|
| Game solution | 128.9 | 132.7 | 128.7 | 129.2 |
| Ideal case | 80 | 82.4 | 135.1 | 137 |
| Random case | 207.9 | 209.6 | 524.5 | 537.4 |

## V. CONCLUSION

We developed a game engine for spectrum access of PUs and SUs in the presence of cognitive interferers. The game engine is based regret minimization for channel selection and power control. There exist several forms of uncertainty, including channel conditions, packet traffic and availability of users across channels, further aggravated by the receiver link delays (such as in a SATCOM system) that make spectrum efficient communication a challenging task. To better capture the dynamic nature of random utility functions, we applied regret minimization that provides low complexity and fast solution comparing to traditional game solutions based on expected utility maximization. A separate game engine was run at each node independently and a suitable policy was selected. This approach has low complexity and fast convergence speed. Effects of spectrum sensing on throughput and delay were studied. Results were also compared to the ideal case and the random case where users have perfect or no knowledge on channel availability, respectively.

## REFERENCES

[1] A. Blum and Y. Mansour, "Learning, regret minimization, and equilibria," Chapter 4 in *Algorithmic Game Theory*, N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, eds. Cambridge University Press, 2007.

[2] A. B. MacKenzie and L. DaSilva, *Game Theory for Wireless Engineers*, Morgan & Claypool Publishers, 2006.

[3] M. Bennis, S. Perlaza, and M. Debbah, "Learning coarse-correlated equilibria in two-tier networks," in Proc. IEEE International Conference on Communications (ICC), Ottawa, Canada, June 10-15, 2012.

[4] M. Dinitz, "Distributed algorithms for approximating wireless network capacity," in Proc. *IEEE INFOCOM*, San Diego, CA, March 15-19, 2010.

[5] A. Anandkumar, N. Michael, A. K. Tang, and A. Swami, "Distributed algorithms for learning and cognitive medium access with logarithmic regret," *IEEE Journal on Selected Areas in Communications*, vol. 29, pp. 731-745, April 2011.

[6] Z. Han, C. Pandana, and K. J. R. Liu, "Distributed opportunistic spectrum access for cognitive radio using correlated equilibrium and no-regret learning," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, Hong Kong, China, March 11-15, 2007.

[7] N. Nie and C.Comanciu, "Adaptive Channel Allocation Spectrum Etiquette for Cognitive Radio Networks," Mob. Netw. Appl., 2006.

[8] Q. Zhu and T. Basar, "No-regret learning in collaborative spectrum sensing with malicious nodes," in Proc. *IEEE International Conference on Communications (ICC)*, Cape Town, South Africa, May 23-27, 2010.

[9] T. Pratt, C.W. Bostian, and J. Allnutt, *Satellite Communications*, Second Edition (John Wiley & Sons), 2003.