A Secure Radio Environment Map Database to Share Spectrum

Shabnam Sodagari, *Senior Member, IEEE*

***Abstract*—A robust and secure database for spectrum sharing in cognitive radio networks that is obscured from the viewpoint of secondary users is presented. The database allocations secure features of white space resource usage from being learned. The design of non-inferable database is based on two cases. In the first case, the primary or spectrum lender has no knowledge of secondary users or potential jammers among them. In this case, the problem is modeled as a Markov decision process. In the second case, the primary system has some knowledge about spectrum borrowers and the problem is modeled as a Bayesian Stackelberg game. Mutual information interpretation of the Bayesian Stackelberg game is also presented. The solutions facilitate releasing more bandwidth as required by US National Broadband Plan. Applications of this scheme are manifold. This design can be used for securing spectrum resources, for example radar white spaces, while being shared with LTE and commercial communication systems. Further, it provides jammer-proof spectrum sharing among various communication, detection, and navigation systems. Simulation results verify this scheme improves system throughput while maintaining desired obfuscation level or entropy.**

***Index Terms*—Cognitive radio (CR), dynamic spectrum access, radio environment map database, security.**

I. Introduction

COGNITIVE radio (CR) entered the lexicon of wireless communication as a means to enable dynamic spectrum access to increase spectral efficiency in wireless systems. In a cognitive radio network (CRN) the spectrum license holder is called a primary user (PU). Other devices that try to dynamically access the unused resources of PU, without affecting its performance, are called secondary users (SUs) or CRs. We interchangeably use the terms secondary user and CR.

The traffic pattern of PUs is varying, because PUs are either busy, i.e., using the link, or idle. In addition, due to inherent lower priority of CRs, they should adjust their transmission parameters to comply with PU interference requirements. There are two main approaches toward this requirement. First, CRs can sense the spectral resources of primary system and start transmitting when they find those to be idle, i.e., not being utilized by PU. However, spectrum sensing results may be inaccurate due to shadowing and fading effects in wireless channels. In the second approach, the PU plays an active role and provides a database containing information of its spare resources. In this case, the database can further include a policy reasoner to manage allocation of spare resources to secondary users. Throughout this paper, the terms database and policy reasoner are used interchangeably.

A. Problem Statement

Security is a major barrier that discourages both commercial and federal applications, including radar, to be willing to share their spectral/temporal and spatial resources with other communication, detection, and navigation systems, such as LTE or WiMAX. For example, classified features, e.g., bandwidth, central frequency, azimuth, etc of radars may be revealed or learned by the secondary system, such as commercial communication users, during sharing. This is due to the risk of potential jammers. During the sharing process, jammers might be able to establish a learning mechanism to find out what parameters in these domains are being used and to what extent. Therefore, to solve this problem the challenge is finding methods for securing the features of white space resource usage and openings from being learned by an outside observer, while at the same time enabling the spectral, temporal, and spatial resources to be shared with commercial communication systems. This is the holy grail to realize the National Broadband Plan. In order to solve this problem, several challenges need to be overcome:

- Challenge 1: The non-learnable allocation and scheduling module should guarantee to use resources optimally.
- Challenge 2: The obscured allocation and scheduling module should take into account uncertainties in the information about the radio environment map (REM).
- Challenge 3: The obscured allocation and scheduling module should take into consideration the needs input by the commercial communication side as much as possible.
- Challenge 4: The complexity of computations for the obscured allocation and scheduling module should be low to allow real time implementations and update of policy, as the REM changes.

This paper explains a solution that makes spectrum usage patterns obfuscated, for security and public safety reasons, while benefiting commercial consumers of bandwidth, through sharing spectral resources with them. This is a breakthrough, because once implemented it provides incentives for releasing a considerable amount of bandwidth for sharing.

The organization of this paper is as follows. Section II reviews relevant work. In Section III we present the system model and the solution to the problem. Simulation results are presented in Section IV and Section V concludes.

Manuscript received October 20, 2014; revised February 19, 2015; accepted April 18, 2015. Date of publication April 24, 2015; date of current version September 14, 2015. The guest editor coordinating the review of this manuscript and approving it for publication was Prof. Wade Trappe.

The author is with the University of Maryland, College Park, MD 20742 USA (e-mail: shabnam@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2015.2426132

TABLE I
NOTATION AND ABBREVIATION

CR	Cognitive radio
CRN	Cognitive radio network
PU	Primary user
SU	Secondary user
SNR	Signal to noise ratio
DSA	Dynamic spectrum access
MDP	Markov decision process
LP	Linear program
REM	Radio environment map
CRLP	Convex combination for randomization linear programming
BRLP	Binary search for randomization linear programming

Table I contains the notation and abbreviations used throughout this paper.

II. Review of Relevant Work

Spectrum sharing is an enabling technology for the United States' National Broadband Plan, which is expected to reallocate the spectrum in the 3500–3650 MHz band. Some of the 500 MHz of extra spectrum to be made available in accordance with this plan is probably going to be provided by license-holder applications. One major source of fixed license spectrum usage is radar. Therefore, enabling radars to share white spaces with commercial communication systems is vital. On the other side, future broadband wireless access systems would fall in the S-band, which according to IEEE standard spans from 2 GHz to 4 GHz. Traditionally, S-band is used by radars for various purposes that can include weather forecast, surveillance, and aviation.

Spectrum sharing can be made possible by using a database in which organizations and agencies could declare regions where other signals may interfere with their own use of spectrum. As such, they are letting other applications know which parts of their spectrum to avoid and which resources can be safely shared [2].

The need for optimal policies for realizing dynamic spectrum access and sharing among different radio platforms is still to be fulfilled. In policy based architecture there is an element named policy reasoner [3], which is the intelligent decision making entity for resource allocation. This is exactly what this paper is targeting, i.e., design of a novel policy reasoner that randomizes and conceals the spectrum usage patterns during the sharing process. To this end, we take advantage of Bayesian Stackelberg game modeling and Markov Decision Processes, as will be explained in details. We will also address computational complexity to come up with fast solutions suitable for real-time applications. The policy reasoner takes care that resource allocation matches to policies.

We are inserting cognition and intelligence in the policy reasoner, to conceal the sharing patterns from the secondary system side. The major benefit of this approach is that we do not force commercial radio manufacturers to embed some sort of fixed regulatory parameters into their products, which in itself makes reconfiguration of radios costly for consumers when policies and parameters change [4], [5]. DARPA's neXt Generation or DARPA XG program [1], [6] contains system guidelines and enabling technologies for dynamic spectrum access. The XG

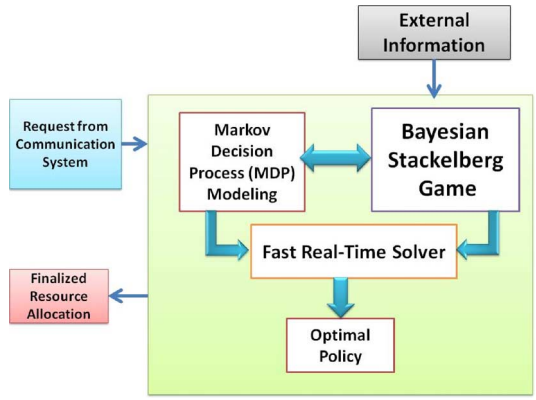

Fig. 1. Block diagram of database policy reasoner for intelligent randomization of primary resources during resource sharing with secondary users.

radio architecture is shown in Fig. 2 [1]. This architecture specifies the role of policy reasoner and its interactions with other cognitive radio elements. This structure offers flexibility and agility for adapting to varying conditions by letting us load policies in a dynamic manner, without requiring extra provisions on radio firmware. The RF component of an XG radio is used to transmit and receive various signals. In order to ensure that the radio's behavior is in accordance with currently active policies and does not cause harmful interference [7], a set of Policy Conformance Components (PCCs) are contained in each XG radio. The major inference and reasoning component of PCC is the policy reasoner. The hardware controlling component is the system strategy reasoner (SSR). The SSR manages the hardware by collecting sensory information and formulating transmission strategies. It is also an interface for transmitting and receiving signals. The SSR interacts with the policy reasoner to determine the available spectrum access opportunities that conform to the currently active set of policies. For example, the SSR formulates a transmission strategy, based on collected sensory information and its current state, and sends this information to the policy reasoner in the form of a transmission request. The policy reasoner assesses the transmission request against the policies to confirm if the transmission strategy is in accordance with the policies. As an example, a mixture of different reasoning techniques, such as partial evaluation, backward chaining, constraint propagation, and forward reasoning have been used in [5] to come up with a suitable policy.

Next, details of applying game and decision theory techniques for obfuscation of spectrum sharing database are explained.

III. System Model and Solution

A. Objectives

We seek solutions to secure the patterns and features of communication, detection, and navigation systems, while they are sharing resources with secondary systems. To this end, we propose to equip the policy reasoner that administers, through a database, the sharing process between the two parties with both a Bayesian Stackelberg game modeling and Markov Decision Process (MDP) tools to randomize the resource allocation and also account for uncertainties in knowledge about the commercial communication users. Fig. 1 shows the functionality of

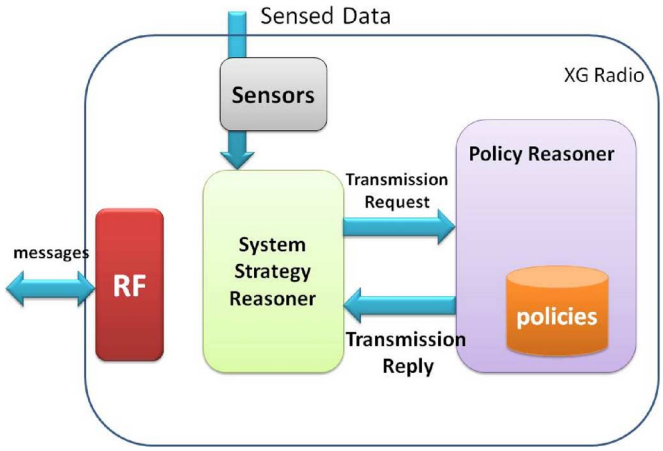

Fig. 2. XG radio architecture [1].

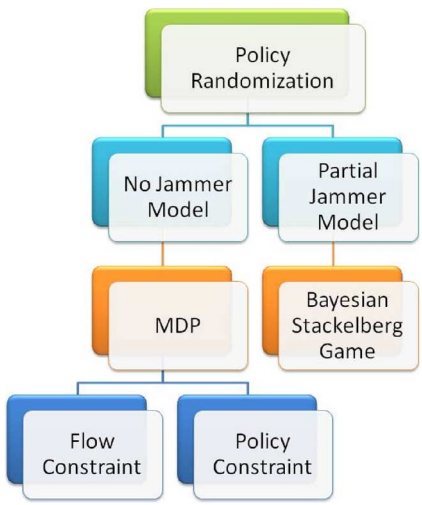

Fig. 3. Hierarchy of the solution for securing spectrum usage and white space patterns.

the database reasoner and Fig. 3 is a snapshot of the solution to insert in DSA architecture for implementing the database reasoner to conceal the resource allocation structure, using randomization, obtained by MDPs and Bayesian Stackelberg game model.

B. Solution

In this section, we explain how we plan to overcome challenges mentioned earlier. So far, randomized solutions based on game and decision theory have not been applied to database obfuscation in DSA and CRN environments. To this end, it is observed that this system shows a similar structure to scheduling of security forces in Los Angeles airport [8]–[10]. There are multiple places in the airport, but the number of security forces to cover those areas is limited. Therefore, security officials in the airport should come up with optimal patrol scheduling policies, within the above mentioned constraints, but in adversarial environments. Hence, patrolling officer scheduling should be randomized in a manner that outside observers cannot learn over time what areas tend to be less covered.

We assume the white space borrower system might include some jammers that can learn about resource allocation of policy reasoner to use it to their advantage. Therefore, the policy reasoner should make the structure of white spaces as imperceptible as possible. When we have no information about jammers, we use MDP to identify randomized solutions that minimize the information exposed to the commercial communication side. In some cases, some limited information about potential jammers exists. For example in sharing with LTE systems, we know that potential jammers have capabilities dictated by LTE consumer profiles. For these cases, we propose using Bayesian-Stackelberg games, considering that we are the lead player. This problem can be NP hard in general. Therefore, some mixed integer programming (MIP)-based solvers like Decomposed Optimal Bayesian Stackelberg Solver and Agent Security via Approximate Policies [8], [9], [11], [12] can be used to make this problem tractable, as applied for securing Los Angeles Airport. Fig. 3 summarizes the essence of the solution for a non-learnable REM database with optimal resource allocation policy [13].

We use two major approaches to tackle the stated problem: game theoretic modeling and decision theoretic models. We then show how it can be related to imperceptible randomization policy. Decision theory and MDPs are useful tools to model stochastic actions when we are making decisions in uncertain environments, e.g., for constraints imposed by REM. However, here, unlike deterministic algorithms that are based on reward maximization [14] for calculating the optimal decision in uncertain conditions, existence of jammers should also be considered. In this regard, policy randomization [15], [16] in adversarial environments [10], [11], [17], [18], where we do not want the decision making process to be learned are investigated.

Stackelberg games modeling is one major framework for security games and modeling two opponents' interactions within the realm of game theory [19], [20]. For example, assuming complete information about adversaries, authors in [21] have developed countermeasures for protecting critical infrastructure against attacks. In this regard, they exploit Stackelberg game models. In real world, the information about adversaries is likely to be incomplete. Bayesian game models and Bayes-Nash equilibrium [22] are useful tools for taking into account incomplete information in a problem [23], [24]. In the following, we explain the above mentioned techniques in more details.

C. Design of Non-Learnable Database With no Knowledge of Jammers

The optimal deterministic policy in a MDP, which is about taking action a in state s, is obtained by [25]

$$\begin{aligned} &\max \sum_{a\in A}\sum_{s\in S} r(s,a)x(s,a) \\ &\text{s.t.} \sum_{a\in A} x(j,a) - \sum_{s\in S}\sum_{a\in A} p(s,a,j)x(s,a) = \alpha_j \ \forall j \in S \\ &\forall a \in A, s \in S, x(s,a) \geq 0. \end{aligned} \tag{1}$$

In (1) S is the set of world states $\{s_1, s_2, \ldots, s_m\}$, A is the set $\{a_1, a_2, \ldots, a_k\}$ of actions, $p(s,a,j)$ is the transition probability of going from state $s \in S$ to state $j \in S$ by taking action

$a \in A$, and $r(s,a)$ is the reward of taking action a at state s. The set of states is the set of resources that can be allocated by database policy reasoner. The set of actions is related to changing allocated resources. For example, states can represent allocated channels. An action taken at a state means transition from allocating one channel to allocating another channel. With α_j denoting the number of times the MDP starts in each state $j \in S$ and $x(s,a)$ being the number of times the MDP visits state s and takes action a, the optimal policy π^* that maximizes the expected reward is obtained by:

$$\pi^*(s,a) = \frac{x^*(s,a)}{\sum_{a' \in A} x^*(s,a')}, \tag{2}$$

where x^* is the solution to (1).

To insert randomness in MDP policy, one can define the weighted entropy function [13], inspired by Shannon entropy [26]. In simple words, the weighted entropy in (3) is defined by taking the sum of the entropy for the distributions at every state weighted by the likelihood the MDP visits that state.

$$\begin{aligned} H_w(x) &= -\sum_{s \in S} \frac{\sum_{a' \in A} x(s,a')}{\sum_{j \in S} \alpha_j} \sum_{a \in A} \pi(s,a) \log \pi(s,a) \\ &= -\frac{1}{\sum_{j \in S} \alpha_j} \sum_{a \in A} \sum_{s \in S} x(s,a) \log \frac{x(s,a)}{\sum_{a' \in A} x(s,a')} \end{aligned} \tag{3}$$

Equation (4) defines the maximum entropy solution for MDP. In (4) there is a constraint on reward threshold $E_{\min}$, which can be adjusted by the user. For example, If we input $E_{\min} = E^*$, where E^* denotes the maximum possible expected throughput, solving (4) yields the maximum expected throughput policy with largest entropy.

$$\begin{aligned} &\max -\frac{1}{\sum_{j \in S} \alpha_j} \sum_{a \in A} \sum_{s \in S} x(s,a) \log \frac{x(s,a)}{\sum_{a' \in A} x(s,a')} \\ &\text{s.t.} \sum_{a \in A} x(j,a) - \sum_{a \in A} \sum_{s \in S} p(s,a,j) x(s,a) = \alpha_j \ \forall j \in S \\ &\sum_{s \in S} \sum_{a \in A} r(s,a) x(s,a) \geq E_{\min} \\ &x(s,a) \geq 0 \ \forall s \in S, \ \forall a \in A \end{aligned} \tag{4}$$

For $E_{\min} = 0$ the above problem finds the maximum weighted entropy policy, without consideration for throughput. As can be seen, there is a tradeoff between reward and randomness. The weighted entropy function is neither convex nor concave in x. This leads to finding other simpler and easily solvable formulations that still take care of both randomness and reward. To this end, consider (5).

$$\begin{aligned} &\max \sum_{a \in A} \sum_{s \in S} r(s,a) x(s,a) \\ &\text{s.t.} \sum_{a \in A} x(j,a) - \sum_{a \in A} \sum_{s \in S} p(s,a,j) x(s,a) = \alpha_j, \ \forall j \in S \\ &x(s,a) \geq \beta \bar{x}(s,a), \ \forall a \in A, \ s \in S \end{aligned} \tag{5}$$

Randomized policy MDPs with a high expected reward solution can be characterized by introducing a randomness indicator variable $\beta \in [0,1]$. Solving equation (5) maximizes the expectation of throughput for a given β and a high entropy solution $\bar{x}$. In other terms, β is the amount of randomness in (5).

For policy reasoner to be able to make real time decisions, (5) should be solved in polynomial time. Two existing polynomial time solutions to (5) are Convex Combination for Randomization Linear Programming (CRLP) and Binary Search for Randomization Linear Programming (BRLP) [13]. These algorithms aim at balancing reward and randomness. The inputs to these two algorithms are a minimal expected reward value $E_{\min}$ (equivalent to throughput) and a randomized solution $\bar{x}$ (or policy $\bar{\pi}$). The role of input $\bar{x}$ is to enforce some level of randomness on the high expected reward output, through linear constraints, and it can be any solution with high entropy. For example, uniform policy $\bar{\pi}(s,a) = 1/|A|$ can be an input to the algorithm. Our objective function is composed of both randomness and reward, and we have to address the tradeoff between these two parameters.

The randomness constraint of the solution is a monotonically increasing function of β. As β increases the expected reward decreases and entropy increases. $\beta = 0$ represents the special case of deterministic MDP as in (1). Therefore, the solution for $\beta = 0$ corresponds to maximum expected reward E^*. When $\beta = 1$ the problem gives the highest possible expected throughput reward $\bar{E}$ only among rewards obtained by entropy dictated by $\bar{x}$. In other words, $\beta = 0$ corresponds to no obfuscation, i.e., only maximizing SU throughput. At the other extreme, $\beta = 1$ corresponds to highest entropy (obfuscation), deprioritizing SU throughput.

Constraint

$$\sum_{a \in A} x(j,a) - \sum_{a \in A} \sum_{s \in S} p(s,a,j) x(s,a) = \alpha_j, \ \forall j \in S$$

in (5) can be written as

$$\mathbf{M}\mathbf{x} = \boldsymbol{\alpha},$$

where $\mathbf{M}$ is a $|S| \times |S||A|$ matrix, $\mathbf{x}$ is a $|A||S|$ dimensional vector, and $\boldsymbol{\alpha}$ is a $|S|$ dimensional vector.

The difference between two algorithms of CRLP and BRLP [10] stems from the fact that the latter is based on reformulating (5) by replacing the flow constraints by policy constraints at each stage, as in (6).

$$\begin{aligned} &\max \sum_{s \in S} \sum_{a \in A} r(s,a) x(s,a) \\ &\text{s.t.} \sum_{a \in A} x(j,a) - \sum_{s \in S} \sum_{a \in A} p(s,a,j) x(s,a) = \alpha_j \ \forall j \in S \\ &x(s,a) \geq \beta \tilde{\pi}(s,a) \sum_{b \in A} x(s,b), \ \forall s \in S, a \in A \end{aligned} \tag{6}$$

By constraining policy $\bar{\pi}$ in (6) instead of the constraint of at least $\beta \sum_{a \in A} \bar{x}(s,a)$ flows reaching each state $s \in S$, one can make sure that the obtained obfuscation is not restricted.

The two following theorems justify CRLP and BRLP polynomial time algorithms.

Theorem 1: If $\bar{x}$ is a feasible solution to problem (1), then $\bar{x}$ is an optimal solution to problem (5) when $\beta = 1$, yielding reward $\bar{E} = \sum_{s\in S}\sum_{a\in A} r(s,a)\bar{x}(s,a)$.

Proof: For brevity, the proof is not presented. The interested reader is referred to [13]. □

Theorem 2: Consider a solution $\bar{x}$, which satisfies $M\bar{x} = \alpha$ and $\bar{x} \geq 0$. Let x^* be the solution to (1) and $\beta \in [0,1]$. If x_β is the solution to (1), then $x_\beta = (1-\beta)x^* + \beta\bar{x}$.

Proof: The proof is based on defining a slack variable for inequality constraint of (5). For brevity, details of the proof are omitted. The interested reader is referred to [13]. □

According to Theorem 2 solution to problem (5) is a convex combination of the deterministic and random input solutions. Theorem 2 also implies the relationship between x_β and β is linear and $\beta = r^T x^*/(r^T x^* - r^T\bar{x})$ gives $E_{min} = r^T x_\beta$. This leads to the following solution, which has tractable computational complexity, making it suitable for real time decision making of resource allocation by the database.

Algorithm 1 CRLP [10]

Take E_{min} and $\bar{x}$ as input
Obtain optimal solution to (1) and call it x^*
Set $\beta = r^T x^*/(r^T x^* - r^T\bar{x})$
Set $x_\beta = (1-\beta)x^* + \beta\bar{x}$
return x_β

In Algorithm 1 r is a $|S||A|$ dimensional vector of rewards and x^* is a $|S||A|$ dimensional vector. For β at values 0 and 1, optimizations (5) and (6) yield the same solution if policy $\bar{\pi}$ is the policy obtained from the flow function $\bar{x}$. However, in the intermediate range of 0 to 1 for β, the policy obtained by equations (5) and (6) are different even if $\bar{\pi}$ is obtained from $\bar{x}$. Thus, theorem 2 holds for (5) but not for (6). For solving (6) BRLP algorithm is used. This algorithm also takes values of minimum reward and a minimum obfuscation level (entropy) as input and iteratively solves (6). In this algorithm, value of β is initialized as 0.5. The algorithm continues by setting β as average of a lower bound and an upper bound. Initial values of lower and upper bound for randomization coefficient β are 0 and 1, respectively. If throughput obtained by solving (6) using β is greater than the input minimum reward value by a threshold, BRLP algorithm updates the lower bound for randomization coefficient to be equal to β. Otherwise, it updates the upper bound to be equal to β and, in the next iteration, updated β is average of updated lower and upper bounds. Iterations are stopped when the difference between throughput and user input becomes less than the threshold.

If the policy reasoner is confident that a specific white space $s \in S$ is not under risk of attack, then the database can set the entropy for that state to 0.

D. Design of Non-Learnable Database With Partial Information about Jammers

In this case, deriving the randomized policy is based on Bayesian-Stackelberg games. In contrast to Nash equilibrium that assumes a simultaneous choice of strategies, fortunately, here, the policy reasoner on the primary cognitive engine is the leader. First, the policy reasoner initiates a strategy of resource allocation decision as the leader. Potential jammers' actions are based on observing the policy reasoner's strategies, i.e., action chosen by the leader. Followers (jammers) optimize their utilities in a selfish manner after observing policy reasoner's resource allocation decision.

Authors in [24] investigate the problem of choosing an optimal strategy for the leader to adapt to in a Stackelberg game, which is NP-hard in the case of a Bayesian game with multiple types of followers. Also, their multiple linear programs (LPs) method involves solving many linear programs. Some of the many linear programs may even be infeasible. According to theorem 2 of [24] finding an optimal pure strategy to commit to, in a 2-player Bayesian game, is NP-hard, even when the follower has only a single type. For proof interested reader may refer to [24]. Hence, authors in [27] resort to methods for finding optimal leader strategies for non-Bayesian games by using Harsanyi transformation to reformulate the Bayesian game into a normal-form game. However, one disadvantage associated with this transformation is losing the compact structure of the Bayesian game. Some methods using mixed-integer linear programs (MILPs) [28] compute the highest-reward Nash equilibrium. This is owing to the fact that the highest-reward Bayes-Nash equilibrium is equivalent to the corresponding Nash equilibrium in the transformed game.

There are both exact and approximate solutions to such games. In contrast to multiple linear programs method [24], the decomposed optimal Bayesian Stackelberg solver provides an exact solution for the choice of optimal policy reasoner strategies, by only solving one LP [8]. This solver does not search for Nash or Bayes-Nash equilibrium. Instead, it searches for optimal high reward non-equilibrium strategies. If the followers act independently, the leader strategy can be decomposed and evaluated against each follower and in the next step, with no need to converting the game to a normal form one (e.g., Harsanyi transformation), the solver expresses the Bayes-Nash game in a compact form.

Denote by x and q the non-learnable database and SU policies, respectively. x consists of a vector of pure strategies of resource allocation of the non-learnable database. The value x_i is the proportion of times in which pure strategy i is used in the policy. Let X and Q denote the index sets of the leader and follower's pure strategies, respectively. The payoff matrices R and C are defined such that R_{ij} is the reward of the leader PU and C_{ij} is the reward of the follower (jammer) when the database takes pure strategy i and the jammer takes pure strategy j. Fixing the policy of the database to a policy x, gives the LP optimization problem as in (7), which the follower solves to find its optimal response to database policy x.

$$\begin{aligned} &\max_{q} \sum_{j\in Q}\sum_{i\in X} C_{ij}x_i q_j \\ &\text{s.t.} \sum_{j\in Q} q_j = 1 \\ &q \geq 0 \end{aligned} \tag{7}$$

Denoting the jammer's optimal response to database strategy x, by vector $q(x)$, the leader (database policy reasoner) maximizes its payoff by the following optimization:

$$\begin{aligned}
&\max_{x,q} \sum_{i\in X}\sum_{j\in Q} R_{ij}x_i q_j \\
&\text{s.t.} \sum_{i\in X} x_i = 1 \\
&\sum_{j\in Q} q_j = 1 \\
&x_i \in [0,\ldots,1] \\
&q_j \in \{0,1\} \qquad (8)
\end{aligned}$$

Constraints related to a feasible policy for the database are the first and third ones. Constraints of a feasible pure strategy for the jammer are the second and fourth. When the database is dealing with multiple resource borrower secondary systems, among which some jammers may exist, denote by x the vector of strategies of the leader and q^l the vector of strategies of secondary system l, with L denoting the index set of SU system types. Similar to the single SU case, X and Q represent the index sets of database and SU system l's pure strategies, respectively. The payoff matrices related to each SU system l, would be R^l and C^l. Given *a priori* probabilities p^l, with $l \in L$, of facing each SU system, the database can decompose the optimization into:

$$\begin{aligned}
&\max_{x,q} \sum_{l\in L}\sum_{i\in X}\sum_{j\in Q} p^l R^l_{ij}x_i q^l_j \\
&\text{s.t.} \sum_{i\in X} x_i = 1 \\
&\sum_{j\in Q} q^l_j = 1 \\
&x_i \in [0,\ldots,1] \\
&q^l_j \in \{0,1\}. \qquad (9)
\end{aligned}$$

E. Mutual Information Stackelberg Game

The goal of obfuscating actions of PU can translate into minimizing mutual information between the database content and the state of PU. When the database allocates some white spaces to a SU, or when it does not fulfill request of SU for a resource, it is inevitably revealing some information to SU about the set of resources generally available to PU. In the simple case of only one SU, if the database always allocates open channels to the SU, then the SU will know the precise allocation of PU spectrum. Resource allocations to SU may have high entropy, but when conditioned on the knowledge that all open channels are allocated to SUs, the mutual information between the database and PU activities becomes maximum. In this case, the jammer gains knowledge of what resources are available to PU.

Therefore, another perspective to look at this problem is to consider a mutual information Stackelberg game. To reduce mutual information between PU activity and database allocations, policy reasoner must make sure its allocations to a particular SU do not span entire set of PU white space. This strategy limits choice of allocations to a SU to a subset of PU resources, but decreases mutual information between PU activity and database allocations, from the viewpoint of SU. For clarity, consider Fig. 4.

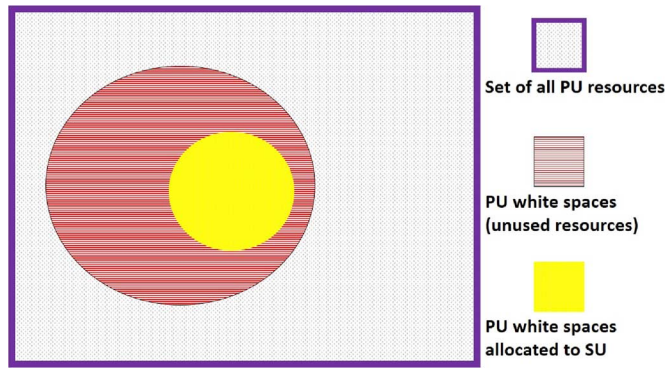

Fig. 4. Venn diagram of strategy for reducing mutual information between PU activities and database allocations.

To decrease mutual information between PU activities and allocated resources to SU, we suggest that the database allocations to a single SU do not span the whole set of available white spaces. This can raise interesting problems to consider. For example, if multiple SUs collude and inform each other of their allocations, they can jointly build an understanding of the entire set of PU resources. Studying this and similar problems are beyond the scope of this paper. Obviously, when the two sets of PU white spaces and PU white spaces allocated to SU, in Fig. 4, are empty, one obtains the extreme case of PU performance maximization, without consideration for obfuscation. Another extreme case is obtained when, in Fig. 4, the set of PU white spaces equals set of all PU resources. Considering that database reasoner is a leader and SU is a follower, SU and database can enter a Stackelberg game in which database tries to minimize mutual information between its decisions and PU usage of resources. On the other hand, SU tries to maximize the above mentioned mutual information by intelligently selecting its submitted resource request to the database. This mutual information game can be zero-sum. Detailed analysis of this scenario is a subject of further research.

IV. Performance Evaluation

A. No Knowledge About Jammers

Simulations were performed, using cvx toolbox for Matlab [29] and IBM CPLEX, for number of channels (or white spaces) representing states varying between 3 to 10 with the number of actions at each state varying between 1 to 8, respectively. Rewards of taking action a at state s were generated randomly as logarithm base 2 of SNRs varying from 10 to 20 dB to represent throughput. Rewards were averaged for 100 realizations. Without loss of generality, possible reward functions can be parameters such as beamforming gain, diversity degree, or ratio of mainlobe to first sidelobe level in angular directions, for example, for radar white spaces. Furthermore, white spaces span a broad possibilities, including but not limited to, frequency channels, spatial/temporal resources, and angular directions.

Fig. 5 shows amount of randomness or entropy obtained by using CRLP algorithm compared with simply allocating the channels in a uniform manner. In Fig. 5 the horizontal axis represents number of channels, which varies between 3 to 10. This algorithm gives almost twice obfuscation in comparison with uniformly distributed resource allocation.

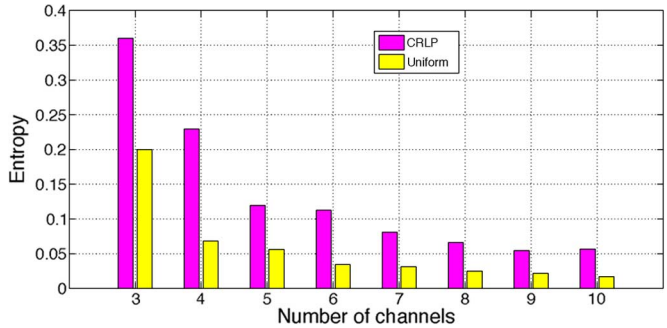

Fig. 5. Comparison of database obfuscation (entropy) obtained by using CRLP and uniform strategy vs. number of channels.

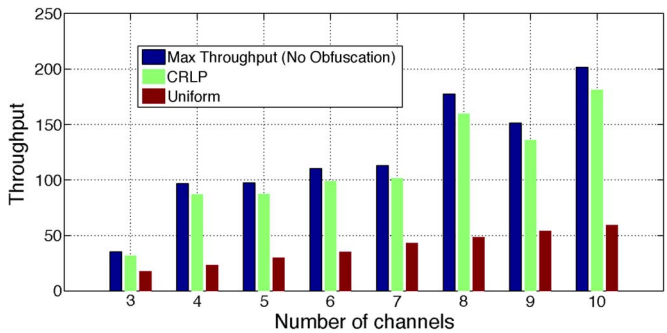

Fig. 6. Comparison of CRLP throughput with throughput of uniform strategy and maximum throughput (no obfuscation) vs. number of channels.

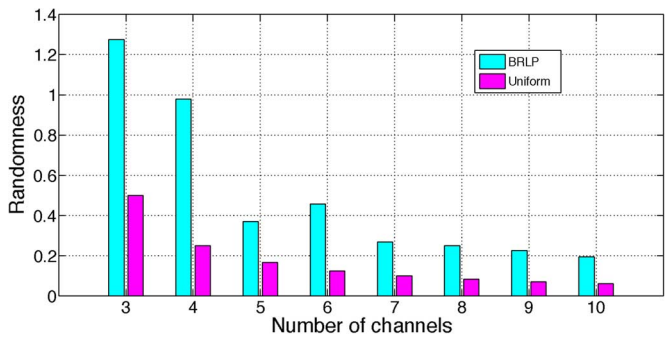

Fig. 7. Comparison of database obfuscation (entropy) using BRLP and uniform strategy vs. number of channels and actions.

Fig. 6 compares overall throughput reward obtained by using CRLP algorithm with maximum throughput strategy, which selects channels with better SNRs and with simply using the uniform strategy. As can be seen, this method is more robust against jamming by yielding more than twice the throughput obtained by uniform strategy, especially as number of white spaces (channels) increases. At the same time, throughput of this method is not very different than the baseline case that allocates channels based on maximum throughput, without obfuscation.

Fig. 7 demonstrates obfuscation level of resource allocation database, expressed as entropy, obtained by using BRLP algorithm to solve (6) and using uniform strategy. Fig. 8 compares the throughput (reward) obtained by optimal randomized resource allocation database using BRLP with the baseline case of allocating for maximum throughput, based on channel SNR, and also with allocating channels in a uniformly random manner. While results are not very different from maximum throughput strategy, they tremendously outperform uniformly distributed randomization.

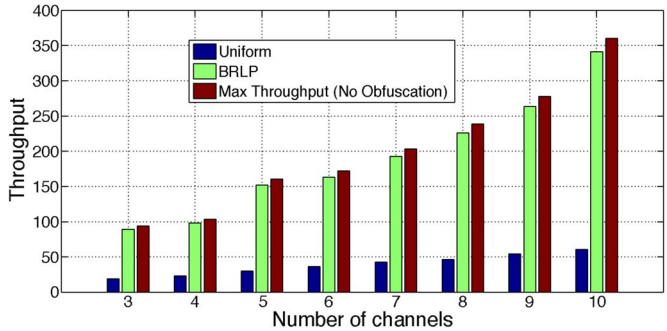

Fig. 8. Comparison of BRLP throughput with uniform and maximum throughput (no obfuscation) vs. number of channels.

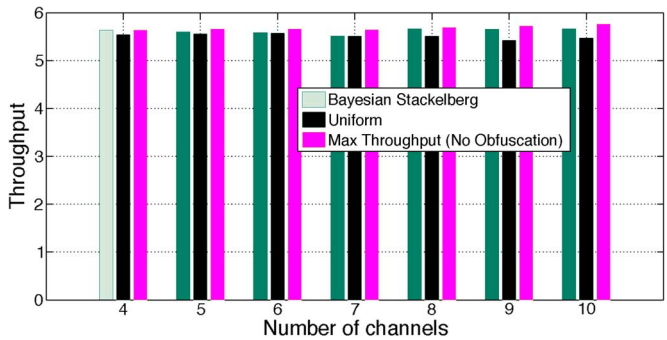

Fig. 9. Comparison of throughput of Bayesian Stackelberg game modeling with uniform and maximum throughput (no obfuscation) strategies.

B. Partial Knowledge About Jammers

We performed experiments with the policy reasoner selecting certain number of subbands to be shared. These may be free subbands or spatial and temporal resources. The Bayesian-Stackelberg game model consists of two players: the policy reasoner and the jammer. The set of pure strategies of policy reasoner consists of a set of white spaces to select and share with SUs such that the SU is not able to learn the structure of availability of primary resources. The policy reasoner can choose a mixed strategy so that the potential jammer will be uncertain of exactly what subbands may be available, but the jammer will know the mixed strategy the policy reasoner has chosen. With this knowledge, the jammer is capable to jam a single subband. If the subband selected by the jammer is not assigned by the database policy reasoner, then the jammer successfully jams that channel.

The payoffs are modeled as capacity in bits/s/Hz with normalized bandwidth, i.e., $B = 1$. Primary system and jammer each have valuations of each channel based on the capacity or throughput of that channel. Jammers may have different valuations for different channels and different costs of getting caught.

The database policy reasoner's set of possible pure strategies or channel selection and allocation can be the set comprising of single channels, or the set comprising tuples of possible combinations of two or more channels. Simulations in this section were performed with the set of single channels as the database pure strategy. The jammer's set of possible pure strategies or subbands to select is denoted by Q and includes integers contained in Q.

Fig. 9 compares throughput obtained by channel allocation based on Bayesian Stackelberg game model with the extreme case of only maximizing throughput, without obfuscation, by allocating higher quality channels, and also allocating channels in

a uniformly random manner. Results were averaged over 100 realizations of reward matrix corresponding to channels' throughputs. As Fig. 9 shows this approach always outperforms uniform channel allocation strategy, while closely following maximum throughput.

V. Conclusion

A non-learnable database was developed for resource sharing. This provides jammer-proof sharing among various communication, detection, and navigation systems. Depending on the knowledge of SU system by PU to which the database belongs, two methods were used. With no knowledge about secondary system, the database decision making process is based on MDPs. When there is some knowledge of SUs, it was shown how the primary system database can be equipped with compact technique for choosing optimal strategies in Bayesian Stackelberg games. The solutions can also be used in CRNs to facilitate releasing more bandwidth as required by US National Broadband Plan. We benchmarked the system performance in terms of two parameters of throughput and obfuscation quantity or entropy. Simulations verify enhancement of overall system efficiency by balancing the performance and desired obfuscation level.

Acknowledgment

Author would like to thank anonymous reviewers for their help in improving the quality of this manuscript.

References

[1] , B. A. Fette, Ed., *Cognitive Radio Technology*, 2nd ed. Amsterdam, The Netherlands: Elsevier, 2009.

[2] Jun. 2012. [Online]. Available: http://bits.blogs.nytimes.com/2012/05/29/how-spectrum-sharing-would-work/

[3] B. Bahrak, A. Deshpande, M. Whitaker, and J. Park, "BRESAP: A policy reasoner for processing spectrum access policies represented by binary decision diagrams," in *Proc. IEEE Int. Dynamic Spectrum Access Netw. Symp. (DySPAN)*, Apr. 2010, pp. 1–12.

[4] J. M. III, "Cognitive radio: An integrated agent architecture for software defined radio," Ph.D. dissertation, KTH Royal Inst. of Technol., Stockholm, Sweden, 2000.

[5] D. Wilkins, G. Denker, M. Stehr, D. Elenius, and R. Senanayake, "Policy-based cognitive radios," *IEEE Wireless Commun. Mag.*, vol. 14, no. 4, pp. 41–46, Aug. 2007.

[6] D. X. W. Group, The XG Vision, Request for Comments, Jul. 2003, prepared by BBN Technologies.

[7] F. Perich and M. McHenry, "Policy-based spectrum access control for dynamic spectrum access network radios," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, pp. 21–27, 2009.

[8] J. Pita, M. Jain, J. Marecki, F. Ordonez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, "Deployed armor protection: The application of a game theoretic model for security at the Los Angeles international airport," in *Proc. 7th Int. Joint Conf. Auton. Agents Multiagent Syst., Industry Track*, 2008, pp. 125–132.

[9] P. Paruchuri, J. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus, "Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games," in *Proc. 7th Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2008, vol. 2, pp. 895–902.

[10] P. Paruchuri, M. Tambe, F. Ordonez, and S. Kraus, "Security in multiagent systems by policy randomization," in *Proc. 5th Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2006, pp. 273–280.

[11] P. Paruchuri, J. Pearce, M. Tambe, F. Ordonez, and S. Kraus, "An efficient heuristic approach for security against multiple adversaries," in *Proc. 6th Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2007, Article No. 181.

[12] A. Murr, Random Checks, Newsweek National News, [Online]. Available: http://www.newsweek.com/id/43401, Accessed 28 Sep. 2007

[13] P. Paruchuri, J. P. Pearce, J. Marecki, F. O. M. Tambe, and S. Kraus, "Coordinating randomized policies for increasing security of agent systems," *Inf. Technol. Manag.*, vol. 10, pp. 67–79, 2009.

[14] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: Wiley, 1994.

[15] D. Dolgov and E. Durfee, "Approximating optimal policies for agents with limited execution resources," in *Proc. 18th Int. Joint Conf. Artif. Intell., AAAI Press*, 2003, pp. 1107–1112.

[16] P. Paruchuri, M. Tambe, F. Ordonez, and S. Kraus, "Towards a formalization of teamwork with resource constraints," in *Proc. 3rd Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2004, pp. 596–603.

[17] D. Carroll, C. Nguyen, H. Everett, and B. Frederick, Development and testing for physical security robots, 2005 [Online]. Available: http://www.nosc.mil/ robots/pubs/spie5804-63.pdf

[18] R. Beard and T. McLain, "Multiple UAV cooperative search under collision avoidance and limited range communication constraints," in *Proc. 42nd IEEE Conf. Decision Control*, 2003, vol. 1, pp. 25–30.

[19] T. Roughgarden, "Stackelberg scheduling strategies," in *Proc. 33rd Annu. ACM Symp. Theory Comput.*, 2001, pp. 104–113.

[20] H. V. Stackelberg, *Marketform und Gleichgewicht*. ViennaNew York, NY, USA: Springer, 1934.

[21] G. Brown, M. Carlyle, J. Salmeron, and K. Wood, "Defending critical infrastructures," *Interfaces*, vol. 36, no. 6, pp. 530–544, 2006.

[22] D. Fudenberg and J. Tirole, *Game Theory*. Cambridge, MA, USA: MIT Press, 1991.

[23] J. Brynielsson and S. Arnborg, "Bayesian games for threat prediction and situation analysis," in *Proc. 7th Int. Conf. Inf. Fusion*, 2004, vol. 2, pp. 1125–1132.

[24] V. Conitzer and T. Sandholm, "Computing the optimal strategy to commit to," in *Proc. 7th ACM Conf. Electron. Commerce*, 2006, pp. 82–90.

[25] D. Dolgov and E. Durfee, "Constructing optimal policies for agents with constrained architectures," Univ. of Michigan, 2003, Tech. Rep.

[26] C. Shannon, "A mathematical theory of communication," *Bell Labs Tech. J.*, vol. 27, 1948.

[27] J. Harsanyi and R. Selten, "A generalized Nash solution for two person bargaining games with incomplete information," *Manag. Sci.*, vol. 18, no. 5, pp. 80–106, 1972.

[28] T. Sandholm, A. Gilpin, and V. Conitzer, "Mixed-integer programming methods for finding Nash equilibria," in *Proc. 20th Nat. Conf. Artif. Intell.*, 2005, pp. 495–501.

[29] CVX: Matlab Software for Disciplined Convex Programming, Version 2. CVX Research Inc., Aug. 2012 [Online]. Available: http://cvxr.com/cvx

Shabnam Sodagari, photograph and biography not available at the time of publication.