Integration of Spectrum Database and Sensing Results for Hybrid Spectrum Access Systems

Ying Dai and Jie Wu
Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122
Email: {ying.dai, jiewu}@temple.edu

***Abstract*—The database-driven spectrum access system recently attracted increasing amounts of attention. It has more benefits compared to the traditional sensing-based systems. To build a more practical and reliable system, a hybrid sensing-based and database-driven spectrum access system is a promising solution. In this paper, we consider the integration problem of the database information and sensing results, which is a very important factor in order in realizing the hybrid system. We propose the integration framework, which is implemented on the database engine. The framework is divided into two main components. The first one is to process the sensing results, which contains the predictions for locations without sensing results, and the fusion policy on the sensing samples. The second component is the dynamic integration process of the generated sensing results and the database information. We first model the evaluation of the integration results as a Partially Observable Markov Decision Process (POMDP), which enables the database engine to know its current status. Then, we propose an iterative algorithm for the database engine to dynamically adjust its integration policy. In this way, the balanced status of the generated spectrum map is maintained. Simulations are conducted to reveal the performance of our framework.**

Index Terms*—*Dynamic spectrum access, database-assisted, hybrid spectrum access systems, integration policy.

I. INTRODUCTION

To solve the spectrum congestion problem, spectrum sharing between primary users (PUs) and secondary users (SUs) is a promising solution [1], [2]. In order to make the spectrum sharing more practical, the key issue lies in the protection of PUs, which requires each SU, or node, to make sure that no active PU exists when accessing the spectrum. Therefore, an effective spectrum access system is necessary.

Many existing works are built on the spectrum sensing-based systems [3]–[5], which have SUs to sense the spectrum and only access it when PUs are inactive. However, it has very high requirements on the sensing accuracy. Also, it is time-consuming and degrades the network performances for SUs. Moreover, the sensing results tend to be more aggressive, since it predicts the spectrum opportunities only based on a certain time period. Once the PUs become active, the corresponding SUs need to quit from the spectrum in order to protect PUs.

Nowadays, a new spectrum access system, which is based on the spectrum database, has attracted more and more attentions [6], [7], which is currently applied in TV white space. The spectrum database usually contains static data, e.g., regulatory rules. Given the SUs' location and time information, the database is able to calculate the access rules. However, the existing methods for calculating the spectrum access rules are overly conservative, which reduces the spectrum opportunities for SUs, especially in metropolitan areas.

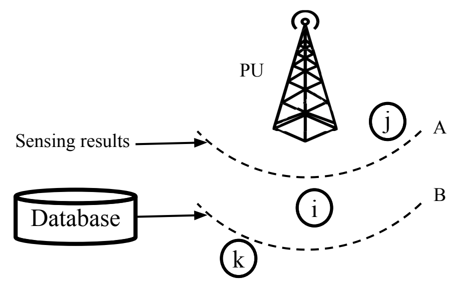


Fig. 1. Motivations for the integration framework.

Given the pros and cons of sensing-based and database-driven spectrum access systems, one obvious solution comes up, which is a hybrid system of both. The information provided by a spectrum database can give the conservative spectrum access schemes, while the SUs' sensing results can be used for local adjustments to provide more spectrum opportunities. However, the hybrid system is not easy to achieve. One question arises: how does the integration of the database information and sensing results work?

To answer the integration question, several issues from different aspects need to be considered. The first one is the coverage issue of sensing results. Since the sensing results are provided by SUs, not every location has the reported sensing results. We need to make use of some statistical methods to predict the sensing results at some locations. The second question is the weighted integration of database information and sensing results. We know that, in general, the database information is conservative and the sensing results are aggressive. But how to quantify the conservative or aggressive levels is a problem. Other issues, like how to format the sensing results to be compatible with database information, are not within the scope of this paper. They are more related to the sensing techniques used and the propagation models of the physical layer.

An example is shown in Fig. 1. Suppose there is a PU and several SUs, $\{i, j, k\}$. When the PU is active, the information given by the database will claim that nodes within boundary B cannot transmit data. However, it is possible that the sensing results of node i and/or other nodes claim that the boundary of the PU area is A. Therefore, for node i, it cannot access the spectrum based on database information while it can see the spectrum opportunity under the sensing results. Therefore, an appropriate integration scheme is necessary. Also, as discussed above, the sensing results cannot cover every location. In this example, we can make use of the sensing results from $\{i, j, k\}$ to make predictions for their nearby locations.

In this paper, we consider the integration problem under the hybrid spectrum access systems of the database-driven and sensing-based models. We assume the existence of the database engine, which takes inputs from database and sensing results, and implements the integration phases. We propose our integration framework, which solves the problem from two components. The first component is to construct the sensing information for locations that have no sensing results reported. The second component is to find a balanced combination of the database information and the sensing results.

For the first component of our framework, we can make use of the spacial-based statistical methods to predict the sensing information for locations that have no sensing results reported. The database engine first collects the sensing results from SUs. It decides the optimal sample size for the fusion of sensing results. Then, it depicts the spectrum map purely based on the collected and predicted sensing information. The constructed spectrum map is not for spectrum access, but for the use of integration. In this way, for any location, both the sensing results and the database information are accessible.

For the second component of our framework, we need to find a balanced combination of the database information and sensing results. Obviously, this is not a one-step work. Instead, we need to evaluate the current weight assignments and make iterative adjustments. To evaluate current settings of the integration, we formulate the process as a Partially Observable Markov Decision Process (POMDP). The database engine needs to first find out if the current combination of database information and sensing results is too aggressive, conservative, or balanced. However, it is difficult for the database engine to tell its current status. The formulation of the POMDP solves the problem for the database engine.

After the database engine evaluates its current status, it needs to make adjust, or simply keep the settings if it is balanced. We propose an algorithm, which uses a stepwise based scheme for weight adaptations. It is performed by the database engine iteratively, combined with the POMDP, to maintain the balanced status of integration policy.

The main contributions of our work can be summarized as follows:

- We propose an integration framework for the hybrid spectrum access systems, which solves the problem of how to combine the static database information and dynamic sensing results.
- We solve the coverage issue of the sensing results by applying spatial statistics-based methods, and make the database engine aware of the optimal sampling sizes for the fusion of sensing results.
- We formulate the evaluation of integration policy as a POMDP, which enables the database engine to become aware of the current status.
- A stepwise-based algorithm is proposed for the database engine to dynamically adjust the weights for integration, which maintains the produced spectrum map in a balanced status.

Our paper proceeds according to the following organization. In Section II, we discuss the related works. We introduce two preliminaries in Section III. The problem formulation is shown in Section IV. We discuss our integration framework in Section V, which provides the details of main components. The performance evaluation is described in Section VI. We conclude our paper in Section VII.

II. Related Works

In this section, we discuss our related works from two aspects. One is about the practical issues for applying geolocation databases on spectrum sharing. Another is about the performance improvement under the spectrum database framework.

A. Practical Issues for Applying Geolocation Database

There have been many works done on the TV white space area. The model "SenseLess" in [7] is under the database-driven white space network. It provides a complete service to ensure an efficient white space networks while protecting PUs. Authors in [8] provide the overview of the radio environment map for the realization of dynamic spectrum access. Their model makes use of multi-domain information from gelocation databases, e.g., characteristics of spectrum use, geographical terrain models, propagation environment, and regulations. The work in [9] compares different estimation methods and studies the accuracy of the signal strength estimation for a primary TV network, which is used for including measurement data into a database's prediction process.

B. Performance Optimization for Database-driven Framework

Many efforts are put into the improvement of the performance for the database-driven spectrum access system. In [10], a game theoretic approach is proposed for the database-assisted white space access point network design. They model the channel selection problem as a distributed game in each access point, and prove the convergence of a state-based Nash equilibrium. Regarding the security aspects, authors in [11], [12] study the location robustness and privacy issues in the database-driven networks. In [13], they focus on improving the accuracy of the geolocation databases through their collected spectrum sensing samples in a TV network area. Since the database requires the SUs' locations to run spectrum information queries, it becomes challenging when SUs are mobile. Authors in [14] propose a framework to support mobile users.

C. Differences of Our Work

Our work proposes a framework to make the application of the geolocation database more practical. We consider both the practical issues and the performance improvement. The differences of our work contain several aspects. Our framework enables the hybrid spectrum access system of both database-driven and sensing-based schemes. Our focus is the integration of the information from both sources and to provide a more reliable and useful spectrum map for each SU. Also, we improve the performance by considering the optimal sample sizes and the dynamic update of the integration policy.

III. Preliminaries

In this section, we discuss two preliminaries for our framework. The first one is the format consistency issue of the database content and sensing data. The second one is the overview of POMDP.

A. Format Consistency

Our framework focuses on the integration process, which defines the fusion policy. However, it is not obvious regarding how to format the sensing information in a consistent manner with the database content. The raw sensing data is very likely to have the different formats as the database records, especially considering when different sensing techniques are used. The formatting process depends on the specific sensing technique used by the SUs [3], [15]. In our paper, we study the scenario of energy detection, and the target data field is the interference to PUs. The raw sensing data contains the received signal strength. Without loss of generality, the collected sensing samples contain two cases, with or without PU signals. There are many existing works [4], [16] regarding the decision policy of sensing results. Moreover, the propagation model plays an important factor during the formatting process, which is highly related to the terrain information. It takes the detected signal strength, terrain information, etc., as inputs, and outputs the estimated detected interference level. In our following sections, we omit the formatting part of the physical layer, and concentrate on the integration process of the conservative and aggressive data.

B. Overview of POMDP

The speciality of POMDP is that instead of tracking one explicit state, it maintains a probability distribution of all possible states, which is a belief state. During the update process, the belief state provides a confidence score. The POMDP framework is very general for modelling a sequential decision process. The optimal action maximizes/minimizes the expected rewards. The factors of a POMDP usually include states, actions, state-transition probabilities reward function, observation, and observation probabilities. We will give more detailed descriptions in our modeling process. The reason we apply POMDP here is because the database engine, which will be introduced later, cannot directly observe its current state of the integration process. The underlying process is a Markov Decision Porcess. The database engine can only maintain a probability distribution over the set of possible states, making observations based on SUs' feedbacks. The modeling and decision process will be introduced in the following sections.

IV. Problem Formulation

In this section, we first describe the network environment and assumptions. Then, we discuss the objectives and challenges of our problem.

A. Network Environment

We assume that there exists the spectrum database, which enables the database-driven spectrum sharing. The database is able to provide the channel availabilities based on static information, such as regulatory rules. The information provided by the database is usually too conservative. It may mark some locations that are capable of dynamic spectrum access as unavailable. The database is centralized, although its architecture can be hierarchical to enable fast access.

Suppose there are a set of SUs, denoted as N, in the area, covered by the centralized database. SUs access the spectrum based on the information of database. Also, SUs perform spectrum sensing, which is independent from database information. But the sensing results are only reported to the database, instead of being used directly by SUs for spectrum access. The sensing results by SUs can be too aggressive, since the sensing results are only valid for a certain time period. Also, it is impractical for the sensing results to cover every location.

The integration of the database information and sensing results is necessary, in order to provide a more accurate spectrum map for dynamic spectrum access. We assume that the database has a database engine, which is able to collect the sensing results from SUs, and has the processing ability to perform integration. Another assumption is that SUs are willing to report their sensing results to the database, to get a better spectrum access performance in the long run.

B. System Objective

The integration results can be viewed as a generated spectrum map at each location. The accuracy of the spectrum map can be evaluated through two metrics: 1) missing detection, p_m; 2) false alarm, p_f. p_m is the probability of missing the detection of an active PU, and p_f is the probability of falsely reporting an available spectrum opportunity as unavailable. Our objective is to minimize $p_f + p_m$ through the integration process. The constraint of our problem is protecting PU sessions from being interfered with.

Due to the coverage and accuracy limitations of database information and sensing results, it poses two main challenges to our model:

1) The sensing results are impractical to cover every location. Therefore, for a certain location, the database is unable to perform the integration process without the sensing results;
2) The database has no knowledge about the conservative level of the database information, or the aggressive level

of the sensing results. It needs to find an approach to combine the results from two resources.

With the above challenges taken into accounts, in the following sections, we propose our solution to achieve the objective. We consider the practical issues as well as the performance requirements, and present our approach from two main components.

V. Integration Framework

In this section, we propose our framework for the integration of database information and sensing results. We first describe our framework overview. Then, we present the two components of our approach in detail.

A. Framework Overview

Our framework can be divided into two main components. One is the generation of sensing information for some locations without their corresponding sensing results reported. Another one is the dynamic combination of the database information and sensing results. The overview of our framework is as follows:

- For a certain location, if there are sensing results reported by SUs, then nothing needs to be done. Otherwise, the database engine needs to apply spatial statistics-based methods, as to generate the corresponding sensing information. Then, the database engine purely uses the sensing information to depict the spectrum map;
- After the spectrum availabilities are generated by sensing results, the database engine combines the results with the corresponding availability information existing in the database. This integration process is formulated as a partially observable Markov decision process. The weights of both resources are adjusted dynamically, based on feedbacks of p_f and p_m values.

We assume that the sensing results reported by SUs are accurate. The security related issues are out of the scope of this paper. When SUs need to access the spectrum for data transmission, they perform spectrum sensing, and report the sensing results to the database. The database performs the integration, and the integrated results would be returned to the SUs. With the spectrum map returned by the database, SUs access the spectrum and return the feedbacks back to the database. The feedbacks contain information regarding whether PUs are found or SUs sessions are interrupted by PUs. We will discuss the details in the following subsections.

B. Depicting Sensing Results

The depiction of the collected sensing results is performed by the database engine. It contains the process of retrieving information from the collected sensing results, which can be further divided into two phases: 1) estimate the sensing information for locations without sensing results reported; 2) process the collected and estimated sensing results to compute the spectrum availability for each location.

Algorithm 1 Depicting sensing results by database engine.

1. **for** each location l_i **do**
2.   **if** s_i is not collected **then**
3.   Calculate s_i using Eq. (1);
4.   Calculate the availability using Theorem 1;
5. **return** The generated spectrum map.

1) Measurement-based estimation: There are many algorithms that can be applied for the measurement-based estimation. We apply the inter- and extrapolation here. This is because, in our model, the local spatial correlation regarding the spectrum sensing results can be found easily.

We adopt the interpolation by Delaunay triangulation. Suppose that the sensing results are expressed by sensed signal strength. For a node i, we use s_i to denote the sensed signal strength by node i, and vector l_i to denote the location of i. $l_i =< lx_i, ly_i >$, where lx_i is the latitudinal direction and ly_i is the longitudinal direction. To predict the sensing results at a given location, we assume that the sensed signal strength at the location of node i is unknown. Then, suppose that node i and three other nodes h, j, k, are within a convex hull, and the sensing results at these three nodes are unknown. Then, s_i can be calculated as:

$$s_i = s_h + \alpha s_j + \beta s_k, \tag{1}$$

where

$$\alpha = \frac{lx_i ly_k - ly_i ly_k}{lx_j ly_k - ly_j lx_k},$$

$$\beta = \frac{lx_i ly_k - ly_i ly_k}{lx_k lx_j - ly_k ly_j}.$$

For scattered signal outside the convex hull, extrapolation can be applied by calculating the gradient of the boundary. The inter- and extrapolation is straightforward for illustrating our framework. Other methods can also be used under different scenarios.

2) Fusion policy: Another problem that arises here is the fusion policy of each node's sensing results. This is important in our model, since we put the calculation at the database engine, and the fusion policy is a key factor for the spectrum database to depict the retrieved sensing results. Traditional AND/OR rules are simple and easy to compute. However, since the database engine can have very powerful calculation ability, we can adopt the optimal voting rule for the fusion of our framework, which is represented as K_{opt}-out-of-K voting rule (the proof can be found in [17]):

Theorem 1. *Suppose the total sample size is K, the optimal voting rule for minimizing $p_f + p_m$ at the database engine is:*

$$K_{opt} = min\ \left(K, \left\lceil \frac{K}{\lambda} \right\rceil \right), \tag{2}$$

where λ is the measurement of each node's sensing ability, and $\lceil \cdot \rceil$ is the ceiling function.

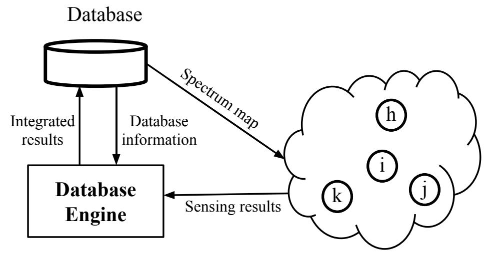


Fig. 2. An example of the integration and spectrum access.

Having both the measurement-based estimation method and the fusion rule ready, the algorithm that runs on the database engine is described in Algorithm 1. After the spectrum availability is generated for each location, the spectrum map can be obtained. One thing to notice is that the information contained in the spectrum map is multi-dimensional, which could contain the location, available time duration, and spectrum frequencies. The information of spectrum map can be altered based on different user queries.

The sensing results are used for integration, rather than the spectrum access at each node. An example is given in Fig. 2 to illustrate the process. Suppose there are four locations of nodes $\{h, i, j, k\}$. The sensing results are collected by the database engine at locations $\{l_h, l_j, l_k\}$, and there are no sensing results at l_i reported. The database will first estimate the spectrum sensing results at l_i, and calculate the spectrum map for the four locations using Algorithm 1. However, the four nodes do not use the sensing results for spectrum access, but use the spectrum map returned by the database. The sensing results are integrated with the database information, and are written to the database by the database engine in this example. Later, if node i sends queries for spectrum access, the spectrum map returned by the database will include the updated results.

C. POMDP Formulation For Integration Assistance

As discussed above, the integration of the database and sensing information needs to minimize $p_f + p_m$. Given the information from two resources, we need to adjust the weights for the values, e.g., availability, signal strength, transmission parameters, and so on, of both resources.

For the database engine, given the current settings of weight distribution, it needs to first decide whether its current combination is balanced, or too conservative or aggressive. However, it cannot directly identify the situation without any measurements or observations. Also, after each measurement, e.g., the feedbacks collected from SUs, the database engine needs to decide whether to continue waiting for more measurements, or to conclude its current state. If it concludes its current state from balanced, then no adjustments on the weights are necessary. Otherwise, an adjustment on the weights is needed, based on whether the current weight settings are conservative or aggressive.

To solve this problem, we formulate it as a POMDP. The following parts specify the ingredients for it, which are states, actions, state-transition probabilities reward function, observation, and observation probabilities.

We denote C as the conservative state, A as the aggressive, and B as the balanced state. A conservative state indicates a large value of p_f, and an aggressive state comes with a large value of p_m. Then we have the following definition for states:

Definition 1. ***States.*** *The possible states in our POMDP is denoted as $\mathcal{S} = \mathcal{E} \cap \{\tau\}$, where $\mathcal{E} = \{C, A, B\}$ and τ is the termination state.*

Note that the termination state means that the databases takes no further measurements.

Definition 2. ***Actions.*** *The actions are denoted by set $\{a\}$. We use $a = 0$ to denote taking a measurement, and $a \in \{1, 2, 3\}$ means to take conclusions and produce the result. We use $a = 1$ to represent the conclusion that the current state is C, $a = 2$ to represent A, and $a = 3$ to represent B.*

Each measurement here means that the database engine collects the feedbacks from SUs for a static period of time. The results produced here are one of states in $\mathcal{E}$.

The state-transition probabilities describe how the state evolves at each step, after taking one action. In our model, even though the database engine takes more measurements, the actual state remains the same. If the database takes no more measurements, then the state changes to the termination state.

Definition 3. ***State-transition probabilities.*** *The state-transition probabilities are:*

$$T(e'|e, a) = \begin{cases} 1 & \textit{if } (a \neq 0, e \in \mathcal{E}, e = e') \\ & \textit{or, } (a = 0, e' = \tau) \\ 0 & otherwise. \end{cases} \tag{3}$$

The reward function is defined based on whether the database identifies its current state correctly. Also, as pointed out before, each measurement takes the time cost of the database engine and the SUs. Therefore, we have the following reward function definition.

Definition 4. *The reward function R is:*

$$R(e, a) = \begin{cases} -\mathcal{M} & \textit{if } a = 0 \textit{ and } e \neq \tau, \\ 1 & \textit{if } a \neq 0 \textit{ and } e = e_a, \\ -1 & otherwise. \end{cases} \tag{4}$$

$\mathcal{M}$ is the average cost for each measurement, e_a is the corresponding concluded state of taking an action a, where $a \in \{1, 2, 3\}$.

Obviously, if the database takes more measurements, the reward relates to the cost. If the conclusion about the current state is correct, the reward is 1. Otherwise, it is -1. Then, the database needs to produce the correct conclusion within the time range, which means the maximum time allowed for measuring before making a conclusion. The overall rewards would be the 1 or -1, which depends on whether the conclusion is correct, minus the total measurement cost. The observations are the measurements based on the feedbacks from SUs. Therefore, we can have the following definition.

Definition 5. ***Observations.*** *The observation space is the set of possible measurements from SUs, denoted as $O = \{o\}$.*

For simplicity, we use the average transmission time over the static measurement period, and the average number of channel switches to denote the measurements. A larger value of either p_f or p_d (conservative or aggressive) will cause the average transmission time to decrease. A larger value of p_d (aggressive) will cause the average number of channel switches to increase.

Definition 6. ***Observation probabilities.*** *The observation probabilities are:*

$$O(e,a,o) = \begin{cases} P(o|e,a) & \text{if } a = 0 \text{ and } e \in \mathcal{E}, \\ 0 & otherwise. \end{cases} \tag{5}$$

If the state is τ, any value of the observation probability does not matter. This is because after the termination τ, the database engine takes no more measurements and the value of the observation probability does not affect the solution. If $a \neq 0$, no measurement is observed. The value of $P(o|e,a)$ denotes the probability of observing o, given last action a and current state e. We set thresholds on the average number of channel switches and the average transmission time. Based on the settings of the thresholds, a larger value of channel switches and a smaller value of transmission time indicates a higher possibility of state A. A smaller value of channel switches and a smaller value of transmission time indicates a higher possibility of state C. Otherwise, it indicates B.

The database engine keeps its belief state b to contain its past information. Given the previous belief state b, the updating of the belief state, b', is based on the last action a, the current observation o:

$$b'(e') = \frac{O(e',a,o)\sum_{e\in\mathcal{S}} T(e'|e,a)b(e)}{\sum_{e'\in\mathcal{S}} O(e',a,o)\sum_{e\in\mathcal{S}} T(e'|e,a)b(e)}. \tag{6}$$

The policy of a POMDP is to map the current belief state into an action. Therefore, the optimal policy should maximize the value function of the belief MDP. Suppose the initial belief state is b_0 and the time horizon is X. Then, the Bellman function is:

$$V_H^*(b_0) = \max_a \left(\sum_{e\in\mathcal{S}} b_0(e)R(e,a) + E\left[V_{H-1}^*(b_1)|b_0,a\right] \right),$$

where b_1 is the belief state after taking action a at b_0, and $E[\cdot]$ is the expectation of $V_{H-1}(b_1)$ given b_0 and a. We use Q to denote the value function of taking action a at state b_x, where $0 \le x \le X$. Then,

$$Q_{X-x}(b_x,a) = \sum_{e\in\mathcal{S}} b_x(e)R(e,a) + E\left[V_{H-x-1}^*(b_{x+1})|b_x,a\right].$$

Algorithm 2 Dynamic integration to calculate $\tilde{I}$.

$status = conservative$; // $status$ has three values.
$unchanged = true$, $step = w/2$;
while $status \neq balanced$ **do**
  if $unchanged = false$ **then**
    $step = step/2$;
  if $status = conservative$ **then**
    $w = w - step$;
  else
    $w = w + step$;
  $\tilde{I} = w \times I_d + (1-w) \times I_s$;
  Update $status$ based on the POMDP output;
  if $status$'s value is changed **then**
    $unchanged = false$;
  else
    $unchanged = true$;
return $\tilde{I}$.

We denote the optimal policy as π^*. Then, given state b_x is, the optimal action $\pi^*(b_x)$ is:

$$\pi^*(b_x) = \arg\max_a Q_{X-x}(b_x,a). \tag{7}$$

Generally, the stationary optimal policy can be achieved if the value of X is sufficiently large. Our focus here is to show how to formulate the problem as a POMDP, and will omit the theoretical approximation parts here. There are many approximation algorithms that can be directly applied, since it is impractical to achieve the stationary optimal policy. In our simulations, we will use the heuristic value iteration algorithm, by setting different upperbound values of X.

D. Dynamic Integration Policy

After the database engine decides whether its current weight distributions are balanced, or too conservative or aggressive, it can dynamically adjust the weight distributions. Here, we apply the stepwise of weight adaptations, which is combined with the POMDP to dynamically integrate the database information and sensing results.

We study the threshold of maximum allowed interference at a given location. The other information stored in the database can be integrated with the sensing results in the similar way. The main idea of this dynamic integration process is very straightforward. We set the weight of the database information as w, with the initial value 1. Then, the weight of the sensing results is $1-w$. For a given location, we use I_d to denote the maximum allowed interference in the database, and I_s to denote the value in the sensing results. How to retrieve the value of I_s from sensing results has been studied a lot. We omit the part here, and will use the average value in a certain

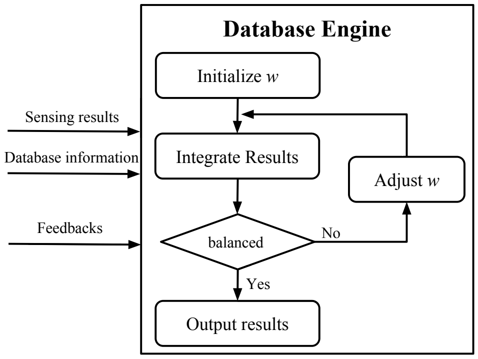


Fig. 3. Main flow graph of database engine.

time duration for simulations. Then, the integration function of the maximum allowed interference at a given location is:

$$\tilde{I} = w \times I_d + (1 - w) \times I_s. \quad (8)$$

By setting the initial value of w as 1, it makes sure that PUs are not initially interfered with. Then, the algorithm is shown in Algorithm 2. The three values of $status$ denote if the current weight settings are balanced, conservative, or aggressive. The process is to dynamically adjust the value of w until the status is balanced. The stepwise is initially large, and would be reduced to reach the balanced status. There are oscillations of the status values. Increasing or decreasing the value of w depends on the current value of status.

Note that Algorithm 2 ends only when the status is balanced. However, to be more practical, we can set an acceptable range to determine if the status is balanced, instead of a specific value. Another consideration is every time we divide the $step$ by 2 to converge to the final value. Other similar methods can be applied to update the value of $step$, as long as the balanced status is achievable. The algorithm is run by the database engine after the POMDP is finished.

Fig. 3 gives an example of the flow process in the database engine. It takes the initial value of w. The inputs are the sensing results and database information. After using the feedbacks, the database engine concludes the status through POMDP. It outputs the results under the current value of w, if the status is balanced. Otherwise, the database engine adjusts w using Algorithm 2 until the balanced status is arrived. The output results would be used as the spectrum map for nodes to access the spectrum.

The integration algorithm is conducted by the cognitive engine, which indicates that we do not need to put too much effort into reducing the complexity. This is because the computing ability of the cognitive engine can be very powerful. Moreover, since the sensing results are dynamic, the database engine needs to trigger the integration process if the current balanced status is broken.

VI. Performance Evaluation

In this section, we first describe our simulation settings. Then, the simulation results are presented.

TABLE I
Simulation settings.

Number of nodes	$[100, 300]$
Number of channels	$[2, 10]$
Average sensing time	$0.5s$
TX power	23 dBm
Noise power	-98 dBm
SINR threshold	10 dB
Number of PUs	$[10, 50]$
PU active duration	$[20, 30]s$
PU active period	$[10, 20]s$
Operation range of each PU	$[300, 500]$

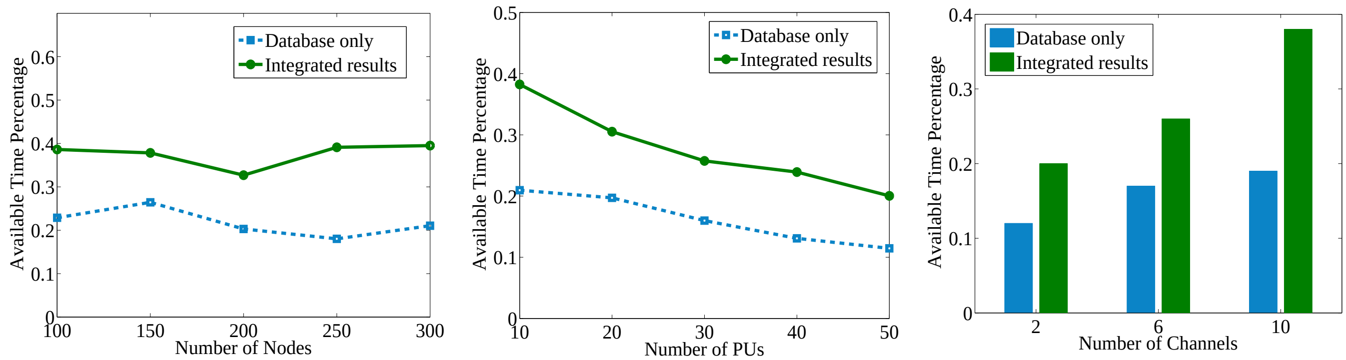
A. Simulation Settings

Table I gives the overview of our simulation settings. Since our model contains several components, e.g., PUs, SUs, and spectrum database, we present the parameters from different aspects. Then, the evaluation metrics will be discussed. Our simulation area is a network with 2000×2000 unit squares. There are a total of 10 channels in the area. PUs and SUs are located in the areas. The spectrum databases store the relevant static information of PUs.

1) PU settings: we randomly distribute a set of PUs in the area. Each PU has its own transmission power and is randomly assigned a channel. We apply the SINR threshold for PUs to calculate their maximal allowable interference at a certain location. Also, each PU is configured with an active pattern, and is periodically turned active for the active duration. Details of the active periods and ranges are shown in Table I. Each PU is randomly set as active at the beginning, and switches between the two statuses.

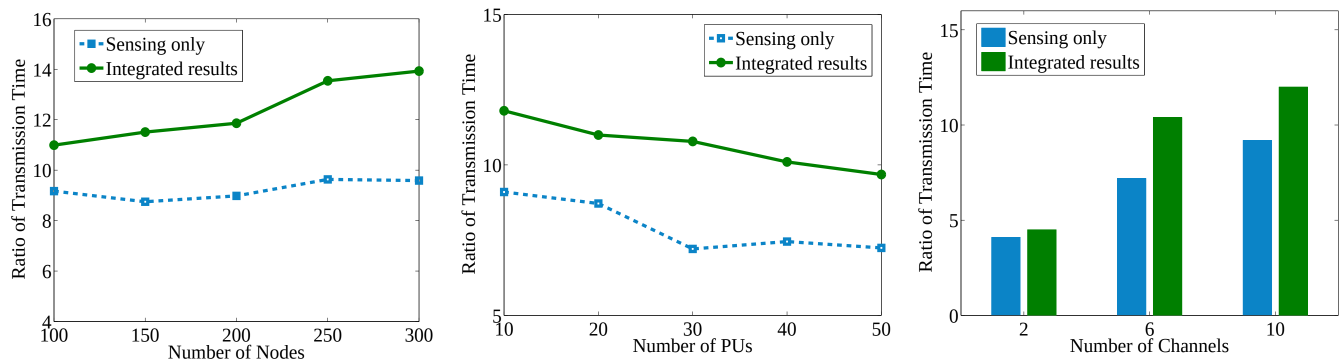
2) SU settings: a number of SU nodes are randomly distributed in the area. Each SU performs spectrum sensing, and collects the sensed PU signal strength at all channels. We omit the sensing technique here by simply using the setting information to provide expected sensing results. We also assume that SU has the SINR threshold information of PUs, and therefore, is able to calculate the maximal allowable interference from its location to PU's transmission range. The calculated maximal allowable interference is sent back to the spectrum database. It is possible that the nearby PUs are inactive when a SU performs spectrum sensing. Therefore, the calculated allowable interference value is too aggressive. Each SU is assigned with a random transmission task at the beginning of each slot. The routing issue is not considered and a SU simply broadcasts the data it has.

3) Database settings: the spectrum information stored in the database conservatively protects the PUs. The database marks the transmission range of each PU, whether it is active or not. The maximal allowable interference stored in the database ensures that the signals from SUs at any location cannot cause the interference within the PU's transmission range. Therefore, the database excludes many spectrum opportunities, regardless of the PUs' status.



(a) (b) (c)

Fig. 4. Comparison of available time percentage under different settings.



(a) (b) (c)

Fig. 5. Comparison of ratio of transmission time under different settings.

4) Evaluation metrics: the evaluation metrics are built from two aspects. One is the extra spectrum opportunities that are created by our hybrid scheme, compared to the scheme only based on spectrum database. The other one is the improved ratio of the transmission time over the sensing time for SUs, compared to the sensing-only spectrum access scheme. After SUs receive the integrated spectrum map from the database, they simply choose the frequency with the longest available duration at their locations. This is a simplified criteria. More advanced criteria contains the scheduling and conflict-avoidance considerations, which are out of our scope in this paper. Therefore, the evaluation metrics of our model are as follows:

- Average available time percentage: we iteratively calculate the percentage of the available time durations over the total time. We calculate the average value of all nodes with transmission tasks in the network for 5 mins. In our model, the percentage is indicated by the integrated spectrum map. We also show the original database spectrum map, which is not combined with the sensing results, for comparison.
- Ratio of transmission time over sensing time: we iteratively calculate the ratio of the transmission time over the sensing time. A larger value of the ratio indicates that the transmission is less frequently interrupted by PUs. SUs need to perform spectrum sensing if they are interrupted by PUs. Otherwise, the transmission continues on the previously found available channels. We use the sensing based scheme for comparison with our model.

Moreover, we vary three network parameters to study the influences on our evaluation metrics. They are the number of nodes, number of PUs, and number of channels. The ranges of their values are also shown in Table I.

B. Simulation Results

We present simulation results based on two evaluation metrics, as described in the above subsection.

1) Available time percentage: We vary the number of nodes from 100 to 300 and calculate the average available time percentage. The number of PUs is kept as 10 and the number of channels is 10. The results are shown in Fig. 4(a). We compare our integrated results after reaching balanced status with the spectrum map, purely relying on the database information. When the number of nodes increases, the average available time percentage of the database-only scheme does not change notably. The value of our integrated results increases slightly. Therefore, the total number of nodes in the network does not have an obvious influence on the average available time percentage. In addition, the integrated spectrum map indicates more available spectrum opportunities than the database-only scheme.

We vary the number of PUs from 10 to 50. The number of nodes is set as 100, and the number of channels is 10. The results are shown in Fig. 4(b). When the number of PUs increases, the values of both schemes decrease. The integrated results outperform the database-only scheme by 30% on the average available time percentage. The gap between the two lines decreases. This is because when the number of PUs increases, there are fewer spectrum opportunities.

The total number of channels varies from 2 to 10 in Fig. 4(c). The number of nodes is 100, and the number of PUs is kept as 10. The available time percentage increases for both schemes when the number of channels increases, since there are more spectrum opportunities. The integrated results still show more spectrum opportunities than the database-only scheme. In addition, the integrated results increase more quickly when the number of channels changes from 6 to 10.

2) Ratio of transmission time: We vary the three network parameters and calculate the ratio of transmission time over sensing time. The results are shown in Fig. 5. We compare the integrated results with the sensing-only scheme. In Fig. 5(a), the number of nodes is varied from 100 to 300. The number of PUs is set as 10 and the number of total channels is also 10. The ratio of the transmission time increases for both schemes. Nodes access the spectrum based on the integrated results, and receive a larger ratio of transmission time over sensing time compared to the ones only based on spectrum sensing.

Next, the number of nodes is kept as 100 and the number of channels is 10. We vary the number of PUs from 10 to 50. The results are shown in Fig. 5(b). The ratio of the transmission time decreases for both schemes. Nodes using the integrated results have a larger ratio of the transmission time over the sensing time.

In Fig. 5(c), the number of channels ranges from 2 to 10. The number of nodes is 100 and the number of PUs is set as 10. The ratio of the transmission time over sensing time increases for both schemes when the number of channels increases. Initially, when the total number of channels is 2, the difference between the two schemes is relatively small. The gap increases when the number of channels increases. This is because when there are few channels, the spectrum opportunities are essentially few.

3) Summary of Simulation Results: In summary, the spectrum map from the integrated results indicates more spectrum opportunities compared to the database contents. Also, nodes who access the spectrum based on the integrated results spend less time on sensing compared to nodes who purely rely on spectrum sensing.

VII. Conclusions

We consider the hybrid spectrum access systems of both the database-driven and sensing-based schemes for dynamic spectrum access. To build a practical and efficient system, we focus on the integration problem of the database contents and sensing data on the database engine. In our integration framework, we first process sensing results, and retrieve the sensing information for locations with or without spectrum sensing reports. Moreover, we describe the dynamic integration process. To make sure the database is aware of its current status, we formulate it as a Partially Observable Markov Decision Process. This enables the database engine to iteratively and dynamically adjust the fusion policy, so that the balanced status is reached. We also propose a straightforward and effective algorithm for the iterative fusion process. In addition, we conduct extensive simulations to study the performances of our integration framework under different network settings.

Acknowledgments

This research was supported in part by NSF grants CNS 149860, CNS 1461932, CNS 1460971, CNS 1439672, CNS 1301774, ECCS 1231461, ECCS 1128209, and CNS 1138963.

References

[1] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, 2006.

[2] Q. Zhao and B. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Processing Magazine*, 2007.

[3] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys Tutorials*, 2009.

[4] "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical Communication*, 2011.

[5] E. Axell, G. Leus, E. Larsson, and H. Poor, "Spectrum sensing for cognitive radio : State-of-the-art and recent advances," *Signal Processing Magazine, IEEE*, 2012.

[6] A. Flores, R. Guerra, E. Knightly, P. Ecclesine, and S. Pandey, "IEEE 802.11af: A standard for TV white space spectrum sharing," *IEEE Communications Magazine*, 2013.

[7] R. Murty, R. Chandra, T. Moscibroda, and P. Bahl, "SenseLess: A database-driven white spaces network," *IEEE Transactions on Mobile Computing*, 2012.

[8] H. B. Yilmaz, T. Tugcu, F. Alagz, and S. Bayhan, "Radio environmentmap as enabler for practical cognitive radio networks," *IEEE Communications Magazine*, 2013.

[9] A. Achtzehn, J. Riihijarvi, and P. Mahonen, "Improving accuracy for TVWS geolocation databases: Results from measurement-driven estimation approaches," in *Proc. of IEEE DySPAN*, 2014.

[10] X. Chen and J. Huang, "Game theoretic analysis of distributed spectrum sharing with database," in *Proc. of IEEE ICDCS*, 2012.

[11] K. Zeng, S. K. Ramesh, and Y. Yang, "Location robustness in database-driven white spaces network," in *Proc. of IEEE DySPAN*, 2014.

[12] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy in database-driven cognitive radio networks: Attacks and countermeasures," in *Proc. of IEEE INFOCOM*, 2013.

[13] J. Ojaniemi, J. Poikonen, and R. Wichman, "Effect of geolocation database update algorithms to the use of TV white spaces," in *Proc. of Crowncom*, 2012.

[14] B. Gao, J.-M. Park, and Y. Yang, "Supporting mobile users in database-driven opportunistic spectrum access," in *Proc. of ACM MobiHoc*, 2014.

[15] A. Ghasemi and E. S. Sousa, "Spectrum sensing in cognitive radio networks: requirements, challenges and design trade-offs," *IEEE Communications Magazine*, 2008.

[16] E. Peh, Y. Liang, Y. Guan, and Y. Zeng, "Cooperative spectrum sensing in cognitive radio networks with weighted decision fusion schemes," *IEEE Transactions on Wireless Communications*, 2010.

[17] W. Zhang, R. Mallik, and K. Letaief, "Optimization of cooperative spectrum sensing with energy detection in cognitive radio networks," *IEEE Transactions on Wireless Communications*, 2009.