

Supporting Mobile Users in Database-Driven Opportunistic Spectrum Access*

Bo Gao
Department of Electrical and
Computer Engineering
Virginia Tech
Blacksburg, Virginia, USA
bgao@vt.edu

Jung-Min “Jerry” Park
Department of Electrical and
Computer Engineering
Virginia Tech
Blacksburg, Virginia, USA
jungmin@vt.edu

Yaling Yang
Department of Electrical and
Computer Engineering
Virginia Tech
Blacksburg, Virginia, USA
yyang8@vt.edu

ABSTRACT

In database-driven opportunistic spectrum access, location information of secondary users plays an important role. In a database query-and-update procedure, a secondary user reports to the geolocation database of its location information, so that the updated knowledgebase facilitates location-aided incumbent protection and network coexistence. However, such database-driven spectrum sharing becomes very challenging when the secondary users are mobile. In this paper, we propose a probabilistic coexistence framework that supports mobile users by incorporating the solutions to solve two core problems: (i) white space allocation (WSA) at the database and (ii) location update control (LUC) at the users. We frame the two problems such that they interact through dynamic control of the users’ location uncertainty levels. For WSA, we derive a centralized real-time solution to mitigate mutual interference among secondary users and protect primary users against harmful interference. For LUC, we design a local two-level strategy to enable both movement-driven and interference-driven control of location uncertainty. This strategy makes an appropriate trade-off between the effectiveness of interference mitigation and the cost of database queries. To evaluate our algorithms, we have carried out both theoretical model-driven and real-world trace-driven simulation experiments. Our simulation results show that the proposed framework can determine and adapt the database query intervals of mobile users to achieve near-optimal interference mitigation with minimal location updates.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design

Keywords

Opportunistic spectrum access; geolocation database; spectrum sharing; mobility management

*This work was partially sponsored by the National Science Foundation through grants CNS-1054697, CNS-1228903, CNS-1314598, and IIP-1265886.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MobiHoc '14, August 11–14, 2014, Philadelphia, PA, USA.
Copyright 2014 ACM 978-1-4503-2620-9/14/08 ...\$15.00.
<http://dx.doi.org/10.1145/2632951.2632984>.

1. INTRODUCTION

In opportunistic spectrum access (OSA), primary users (a.k.a. incumbent users) and secondary users are usually co-located and share the same swaths of spectrum. Incumbent protection (i.e., primary-secondary spectrum sharing) then becomes the major concern, and can be addressed by spectrum sensing techniques and/or geolocation database services. Protecting incumbent users while effectively identifying white spaces is challenging when spectrum sharing relies solely on sensing-only cognitive radios (CRs) [1, 2]. Here, the “white space” refers to fallow spectrum that secondary users can use under the constraint that they do not cause harmful interference to primary users. For this reason, database-driven white space networks [3, 4, 5] have found favor with the spectrum regulators and the wireless industry. Henceforth, we assume a database-driven spectrum sharing system that is consistent with the Spectrum Access System (SAS) for spectrum sharing in the 3.5 GHz band [6, 7], which has recently been proposed by the Federal Communications Commission (FCC). A geolocation database has access to the operating characteristics of incumbent users, such as service types, channel reservations, and protection requirements. In database-driven OSA, a registered secondary user (or its home base station) exchanges information with the geolocation database through a database query-and-update procedure. On the one hand, the user queries the database by its location information and retrieves local white space availability. On the other hand, the user updates the database with its operating information, including its location, so that the updated knowledgebase can facilitate location-aided incumbent protection and network coexistence (i.e., secondary-secondary spectrum sharing) [6, 7, 8, 9].

The above database-driven spectrum sharing system can work well with static secondary networks. However, when the secondary users are *mobile*, the problem of incumbent protection becomes very challenging [4, 10]. Furthermore, the problem of network coexistence, i.e., spatial spectrum reuse and mutual interference mitigation among mobile secondary users, is even more difficult, and has not been addressed in the existing work. When supporting mobile users, each user can query and update the database at a fixed interval, and the *database query interval* (or location update interval) is a key operational parameter that impacts the performance of the spectrum sharing system [7]. On the one hand, during these intervals, the database may have to rely on past location information to ensure incumbent protection and mitigate mutual interference. This leads to considerable loss of white space opportunities [4] and increased likelihood of suboptimal resource allocations. On the other hand, it can be unacceptable to blindly reduce the intervals

because of limited database processing capacity, significant communication overhead rise, and needless network resource waste. This causes the database to become a bottleneck in processing spectrum sharing information. The goal of this paper is to study the strategies for determining and adapting the database query intervals to strike an appropriate trade-off between interference-aware spectrum sharing and cost-effective database access. To the best of our knowledge, this paper is the first work addressing this problem.

We focus on two major problems for supporting secondary user mobility: (i) *white space allocation* (WSA) and (ii) *location update control* (LUC). The database performs WSA and centrally allocates white spaces for secondary users based on their probabilistic *location uncertainty* levels to mitigate co-channel interference among them and ensure no interference with primary users. Each mobile user performs LUC and locally adjusts its location uncertainty level by adapting its database query interval, so that an appropriate trade-off between interference mitigation effectiveness and database query cost can be achieved. The two problems are solved alternately through dynamic control of location uncertainty.

Formulating a solution approach that jointly solves these two problems is challenging. First, the location uncertainty level of a user has to be quantified to model WSA and LUC. Second, the LUC design for each user has to locally adapt database query interval to the time-varying spectrum environment without global knowledge. Third, the WSA design for the database has to promptly adjust resource allocation in response to the dynamic changes in spectrum availability, location uncertainty, and mutual interference probability.

In this paper, we make the following contributions:

- We propose a probabilistic coexistence framework, which incorporates WSA at the database and LUC at the users. This framework jointly addresses the issues of spectrum sharing and database access in database-driven OSA.
- For WSA, we derive a centralized real-time solution that minimizes the probability of mutual interference among secondary users and guarantees full protection of primary users. For LUC, we design a local two-level strategy that minimizes the weighted sum of mutual interference probability and database query frequency (i.e., inverse of the interval). This strategy combines movement-driven and interference-driven control of location uncertainty.
- We evaluate the algorithms for WSA and LUC in two types of simulation experiments. In the first experiment, we have simulated WiFi-like secondary networks using a theoretical mobility model. In the second experiment, we have simulated cellular-like secondary networks using a real-world mobility trace dataset. Our results suggest that the proposed framework determines and adapts the database query intervals of mobile users to realize near-optimal interference mitigation with minimal location updates.
- We provide discussions on system design guidelines and real-world deployment issues in database-driven OSA.

The rest of this paper is organized as follows. Related work is discussed in section 2. System overview is described in section 3. The problems of WSA and LUC are formulated and solved in sections 4 and 5, respectively. Simulation results are shown in section 6. Implementation discussions are presented in section 7, and conclusion is drawn in section 8.

2. RELATED WORK

Recently, the FCC issued a NPRM for enabling small cell use in the 3.5 GHz band [6, 7]. In the proposed framework, spectrum sharing can be achieved among three tiers of users,

including tier-1 incumbent access users, tier-2 priority access users, and tier-3 general authorized access users. The tier-1 and tier-2 users are protected in certain exclusion zones, while tier-3 users have to opportunistically access the band. All the users register with the SAS that incorporates a geolocation database and various interference mitigation techniques. The SAS needs to i) specify appropriate operations across the tiers of users based on location-specific data on spectrum occupancy; ii) resolve interference issues promptly; and iii) coordinate the registered users, if necessary, based on user-generated data on spectrum sharing. The SAS is able to collect location information of registered secondary users and achieve location-aided spectrum sharing.

The research on supporting secondary user mobility in OSA is still in its infancy. The problem of protecting primary users from being interfered with by mobile secondary users has been studied in [4, 10]. In [4], the authors propose a service called SenseLess, which is a database-driven white space network. To enable mobile users without loss of white space opportunities, the database query interval of a user traveling at 60 miles/hour is suggested to be shorter than 30 seconds. However, such a high database query frequency may not always be necessary. When the database manages resource allocation, a certain loss of white spaces is acceptable as long as the remaining white spaces can be enough to offer what the user has requested. For a practical database query strategy, one should consider the trade-off between interference mitigation effectiveness and database query cost. Furthermore, the authors oversimplify the way of setting a database query interval—i.e., they propose that the database query interval of a user is inversely proportional to the speed of the user. For a cost-effective database query strategy, however, one should consider the behaviors of all the primary users and coexisting secondary users in the vicinity to determine the interval. In [10], the authors propose enabling mobile users in a sensing-only CR network. To protect primary users under the location uncertainty of a secondary user, a guard distance is controlled to enlarge each primary exclusion zone for an extra protection. In database-driven spectrum sharing, however, it would be more challenging to achieve the coexistence of mobile secondary users due to the need for dynamic control of both interfering and interfered users' location uncertainty levels.

In CR networks without the issue of location uncertainty, the problems of joint resource allocation have been studied in [11, 12, 13, 14, 15]. In our problem, however, we try to find out how certain a location estimate should be to jointly optimize spectrum sharing and database access.

3. SYSTEM OVERVIEW

In this section, we introduce our coexistence framework and explain basic assumptions and system model.

3.1 Basic Assumptions

We assume a spectrum sharing system similar to that in the 3.5 GHz band [6, 7]. A set of primary users \mathcal{P} and a set of secondary users \mathcal{S} are co-located and share the same set of channels \mathcal{K} . The primary users are protected from any harmful interference in certain pre-defined exclusion zones, while the secondary users have to tolerate harmful interference from others. All the users register with a central database server similar to the SAS in the 3.5 GHz band, which operates a geolocation database with the database query-and-update functionality and performs WSA for the registered users. In this work, the terms “central database server” and “database” will be used interchangeably. The

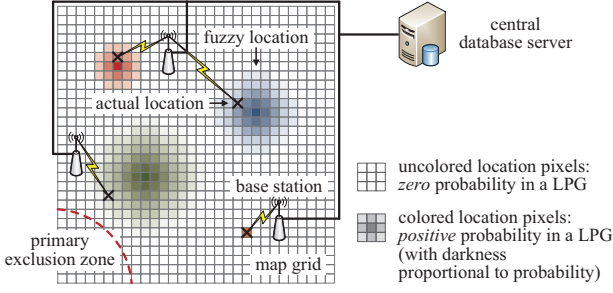


Figure 1: A database-driven spectrum sharing scenario.

base stations offer the interfaces between the registered users and the database. Each user $s \in \mathcal{S}$ can be mobile, and performs LUC for itself. At a certain interval, it updates the database with its current operating information, including its location, and queries the database for location-specific resource allocation. We assume that the users do not share location information among each other for privacy protection and overhead reduction. We assume a time-slot-based system. At the beginning of each time slot, location updates are submitted from the users according to the decisions for LUC in the previous time slot. Using the location updates, the database makes a decision for WSA and notifies the users of changed resource allocation when necessary through their home base stations. Note that even if a user does not query the database, the database still computes resource allocation based on its estimation of the user's location. In this work, a database query is equivalent to a location update. A user accesses the database at most once in each time slot.

3.2 Location Probability Grid

The issues of incumbent protection and spectrum reuse in database-driven spectrum sharing rely on the location information of secondary users. When the users are mobile, their locations can become uncertain and thus affect resource allocation decisions. Hence, a database-driven system needs to quantify the location uncertainty level of a user between successive location update events so that the decisions for WSA and LUC can be adjusted accordingly. In this work, we utilize *location probability grids* (LPGs) to achieve this. In the current designs of geolocation databases, spectrum availability is computed for each location pixel in a pre-defined map grid \mathcal{L} (e.g., a map grid with the granularity of $50\text{m} \times 50\text{m}$ pixel). Hence, as illustrated in Figure 1, we can define a LPG for each mobile user as the probability distribution of the user's location estimate over the map grid \mathcal{L} . For each user $s \in \mathcal{S}$, its LPG is defined by $\mathcal{Q}_s(t) \triangleq \{q_{s,l}(t) \mid l \in \mathcal{L}\}$, where each $q_{s,l}(t)$ is the probability that user s is at location pixel l in time slot t . The LPG of each user defines the fuzziness of its location estimate at the database, which is a function of the elapsed time since the latest location update. The *fuzzy location* (or called movement contour) of user s is defined by $\mathcal{L}_s(t) \triangleq \{l \mid q_{s,l}(t) > 0 \text{ for } l \in \mathcal{L}\}$, which includes all the possible locations of the user in time slot t . Let $\ell_s(t)$ denote the *actual location* of user s in time slot t , which may not be known by the database. The database has to perform WSA based on the knowledge of LPGs. Each user needs to perform LUC to update its LPG when necessary and make its fuzzy location capture its actual location.

3.3 A Coexistence Framework

Intuitively, there exists a trade-off between interference mitigation effectiveness and database query cost. On the one hand, frequent database access decreases the fuzziness

of location estimates and keeps the system deterministic, so that the database can precisely allocate white spaces without losing spectrum opportunities or creating mutual interference. But frequent database access may not always be necessary. For example, when a user occupies a channel exclusively or the user is far away from other co-channel users, fuzzy location of the user is acceptable due to low likelihood of mutual interference. Moreover, too frequent database access may cause delay in server response time or introduce other types of cost such as battery power drain. On the other hand, infrequent database access increases the fuzziness of location estimates, so that large user movement contours may lead to overconservative resource allocation and thus low spectrum utilization. Hence, an optimal database query frequency needs to be found to address the trade-off.

The above trade-off issue involves two problems: (i) WSA, i.e., spectrum sharing controlled by the database; and (ii) LUC, i.e., database access controlled by the mobile users. For WSA, the database first generates LPGs based on the current and past location updates, and then minimizes the probability of mutual interference among secondary users while guaranteeing full protection of primary users based on the updated LPGs. For LUC, the users determine whether to submit location updates next to minimize the weighted sum of mutual interference probability and database query frequency based on the local measurements of user movement and co-channel interference. The problems are solved alternately through the control of LPGs. In the next two sections, the two problems are addressed separately.

4. WHITE SPACE ALLOCATION

In this section, we formulate the problem of WSA and provide a centralized real-time solution.

4.1 Problem Formulation

In each time slot t , the central database server updates its estimation of all the registered users' LPGs based on their current and past location updates. Using the updated LPGs, the database performs WSA to minimize the probability of mutual interference among secondary users under the constraint that the users do not cause harmful interference to primary users. The database achieves this by coordinating the channel assignments and transmit power levels of secondary users. The decision for WSA can be broadcasted to the users through their home base stations.

For WSA that mitigates mutual interference, we need to first create a model of likely interference among secondary users with fuzzy locations. For each user $s \in \mathcal{S}$ and each channel $k \in \mathcal{K}$, define $X_{s,k}(t)$ as a binary channel allocation indicator such that $X_{s,k}(t) = 1$ (0) represents user s is (is not) allocated to channel k in time slot t . Due to spatial spectrum reuse, denote the set of coexisting users that share the same channel k in time slot t by $\mathcal{S}_k(t) \triangleq \{s \mid X_{s,k}(t) = 1 \text{ for } s \in \mathcal{S}\} = \{s_1, \dots, s_{M_k}\}$. A possible location distribution of the co-channel users is $(l_1, \dots, l_{M_k}) \in \mathcal{L}^{M_k}$. Given $\mathcal{Q}_{s_m}(t)$ for all $s_m \in \mathcal{S}_k(t)$, the probability that a particular location distribution (l_1, \dots, l_{M_k}) happens is $\prod_{m=1}^{M_k} q_{s_m, l_m}(t)$. Thus, the database computes the probability of interference experienced at each user s_m on channel k in time slot t by

$$I_{s_m, k}(t) \triangleq \sum_{(l_1, \dots, l_{M_k}) \in \mathcal{L}^{M_k}} J_{s_m | l_1, \dots, l_{M_k}}(t) \prod_{m=1}^{M_k} q_{s_m, l_m}(t), \quad (1)$$

where $J_{s_m | l_1, \dots, l_{M_k}}(t)$ is a binary interference indicator such that $J_{s_m | l_1, \dots, l_{M_k}}(t) = 1$ (0) represents certain intolerable

interference is (is not) experienced at user s_m for location distribution (l_1, \dots, l_{M_k}) . Given a certain (l_1, \dots, l_{M_k}) , the database can utilize a radio wave propagation model to compute the signal-to-interference-plus-noise ratio (SINR) at the receiver of link s_m , denoted by $R_{s_m|l_1, \dots, l_{M_k}}(t)$. Here, link s_m refers to the link in which user s_m can be either a transmitter (in the uplink) or a receiver (in the downlink). We consider basic path loss model as an example, and other models can also be applied in our framework. Let

$$R_{s_m|l_1, \dots, l_{M_k}}(t) = \frac{Y_{s_m, k}(t)H_{l_m, l_m}(t)}{\sum_{\substack{m'=1 \\ m' \neq m}}^{M_k} Y_{s_{m'}, k}(t)H_{l_{m'}, l_m}(t) + Z_0}, \quad (2)$$

where $Y_{s_m, k}(t)$ is transmit power from link s_m on channel k , $H_{l_m, l_m}(t)$ is transmission gain in link s_m , $H_{l_{m'}, l_m}(t)$ is interference gain from link $s_{m'}$ to link s_m , and Z_0 is noise power. Assume a threshold \check{R}_{s_m} such that harmful interference to link s_m occurs if $R_{s_m|l_1, \dots, l_{M_k}}(t) < \check{R}_{s_m}$. Let

$$J_{s_m|l_1, \dots, l_{M_k}}(t) = \mathbf{1} \left(R_{s_m|l_1, \dots, l_{M_k}}(t) < \check{R}_{s_m} \right), \quad (3)$$

where $\mathbf{1}()$ is an indicator function that is equal to 1 (0) if its condition is true (false). Combining (1), (2), and (3), $I_{s, k}(t)$ is a function of $\mathcal{Q}_s(t)$, $X_{s, k}(t)$, $Y_{s, k}(t)$ for $s \in \mathcal{S}$.

Having formulated $I_{s, k}(t)$ under location uncertainty, we can now formulate the problem of WSA. Given $\mathcal{Q}_s(t)$ for all $s \in \mathcal{S}$, the database needs to find $X_{s, k}(t)$ and $Y_{s, k}(t)$ for $s \in \mathcal{S}$, $k \in \mathcal{K}$ to minimize the global objective function

$$I(t) \triangleq \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} I_s(t), \quad (4)$$

where $I_s(t)$ is the average value of $I_{s, k}(t)$ over $K_s(t) \triangleq \sum_{k \in \mathcal{K}} X_{s, k}(t)$ channels taken by user s at a time, namely

$$I_s(t) = \frac{1}{K_s(t)} \sum_{k \in \mathcal{K}} X_{s, k}(t) I_{s, k}(t). \quad (5)$$

There are three constraints for WSA. The first constraint ensures the protection of primary users. In database-driven white space networks, primary users are protected in pre-defined exclusion zones. Hence, all the possible locations of secondary users under location fuzziness must be outside the exclusion zones of co-channel primary users. Suppose each channel k is occupied by a set of primary users $\mathcal{P}_k(t)$ and a set of secondary users $\mathcal{S}_k(t)$. The union of the exclusion zones of primary users in $\mathcal{P}_k(t)$ is defined by $\tilde{X}_{k, l}(t)$, which is a binary channel occupancy indicator such that $\tilde{X}_{k, l}(t) = 1$ (0) represents channel k is not (is) available for secondary users at location pixel l in time slot t . The fuzzy location of each secondary user s is characterized by $\mathcal{L}_s(t)$. Hence, the constraint for incumbent protection is defined as

$$q_{s, l}(t)[1 - (X_{s, k}(t) + \tilde{X}_{k, l}(t))] \geq 0 \quad \text{for } s \in \mathcal{S}, k \in \mathcal{K}, l \in \mathcal{L}. \quad (6)$$

The second constraint avoids secondary user starvation by ensuring a minimum number of channels, \check{K}_s , to be taken by each user s , which is expressed as

$$K_s(t) \geq \check{K}_s \text{ for } s \in \mathcal{S}. \quad (7)$$

The third constraint limits the maximum transmit power, \hat{Y}_s , of each user s , which is expressed as

$$Y_{s, k}(t) \leq X_{s, k}(t) \hat{Y}_s \text{ for } s \in \mathcal{S}, k \in \mathcal{K}. \quad (8)$$

In summary, the problem of WSA solved at the central database server in time slot t can be written as follows.

Algorithm 1 for WSA at the database in time slot t

- 1: **compute** $\mathcal{Q}_s(t)$ for $s \in \mathcal{S}$ according to past location update history submitted from user s
 - 2: **for** each generation **do**
 - 3: **clone** a number of best feasible solutions from previous generation as offsprings
 - 4: **select** enough pairs of good feasible solutions from previous generation as parents
 - 5: **for** each pair of parents **do**
 - 6: **reproduce** a pair of offsprings via crossover, which randomly swaps some rows in matrices $[X_{s, k}(t)]_{|\mathcal{S}| \times |\mathcal{K}|}$ for parent 1 and $[X'_{s, k}(t)]_{|\mathcal{S}| \times |\mathcal{K}|}$ for parent 2 (same crossover is applied for $[Y_{s, k}(t)]_{|\mathcal{S}| \times |\mathcal{K}|}$ and $[Y'_{s, k}(t)]_{|\mathcal{S}| \times |\mathcal{K}|}$)
 - 7: **mutate** each offspring according to a probability, where a row in matrix $[X_{s, k}(t)]_{|\mathcal{S}| \times |\mathcal{K}|}$ is randomly reordered without loss of feasibility (same mutation is applied for $[Y_{s, k}(t)]_{|\mathcal{S}| \times |\mathcal{K}|}$)
 - 8: **end for**
 - 9: **break** when runtime reaches a threshold
 - 10: **end for**
-

Problem 1 (White Space Allocation)

Given: $\mathcal{Q}_s(t)$ for $s \in \mathcal{S}$;

Find: $X_{s, k}(t), Y_{s, k}(t)$ for $s \in \mathcal{S}, k \in \mathcal{K}$;

Minimize: $I(t)$;

Subject to: (6), (7), (8).

Because each $\mathcal{Q}_s(t)$ for computing $I(t)$ follows a general probability distribution, Problem 1 is a mixed-integer nonlinear program (MINLP), which is NP-hard in general. Furthermore, since the database needs to solve Problem 1 in every time slot, it cannot afford the time to find the real optimal solution. Therefore, we extend the genetic algorithm in [11] to derive a fast heuristic solution to Problem 1.

4.2 A Genetic Algorithm

Due to the complexity of Problem 1 and the requirement of fast database response, we apply a variant of genetic algorithm [11] to derive a solution for WSA iteratively in real time. In Algorithm 1, line 1 updates LPGs for computing the objective value, and line 3 and lines 4-8 implement the two stages of cloning and breeding in a genetic algorithm, respectively. In each step of a generation, the candidate solutions are guaranteed to be feasible to Problem 1. A heuristic solution with the best objective value can be obtained when Algorithm 1 terminates. In section 6, we will evaluate the optimality gap of the solution given by Algorithm 1.

5. LOCATION UPDATE CONTROL

In this section, we formulate the problem of LUC and design a local strategy with no need for global knowledge.

5.1 Problem Formulation

In time slot t , each mobile user $s \in \mathcal{S}$ performs LUC to adjust $\mathcal{Q}_s(t+1)$ for the next WSA by determining whether it wants to submit a location update in the next time slot $t+1$. For user s , define $A_s(t)$ as a binary database query indicator such that $A_s(t) = 1$ (0) represents user s submits (does not submit) a location update to the database in time slot t . In general, $\mathcal{Q}_s(t)$ can be generated by a movement prediction model based on the sequence of $A_s(t')$ for $t' = t, t-1, \dots$ and the past location update history in the database.

For LUC that addresses cost-effective database access, we first define the weighted average database query cost (i.e.,

database query frequency) of mobile users in time slot t by

$$C(t) \triangleq \frac{\theta}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} A_s(t), \quad (9)$$

where θ is a positive weight factor, and can be set to reflect various overhead associated with processing query requests at both servers and devices. It may reflect cost factors such as computing power [4], energy consumption, control communication overhead, server response delay, and resource re-allocation frequency. Expression (9) makes the cost function scale proportional to the frequency of database queries.

Having modeled $C(t)$, we can now formulate the problem of LUC, assuming global knowledge of $\mathcal{Q}_s(t)$, $X_{s,k}(t)$, $Y_{s,k}(t)$ for $s \in \mathcal{S}$, $k \in \mathcal{K}$ and future knowledge of $X_{s,k}(t+1)$, $Y_{s,k}(t+1)$ for $s \in \mathcal{S}$, $k \in \mathcal{K}$ at this time. In time slot t , given the solutions to Problem 1 obtained from time slot t and predicted from time slot $t+1$, the mobile users need to find $A_s(t+1)$ for $s \in \mathcal{S}$ to minimize the objective function

$$G(t+1) \triangleq \delta I(t+1) + (1 - \delta)C(t+1), \quad (10)$$

where $I(t+1)$ is the objective function of Problem 1, and $\delta \in [0, 1]$ is a weight factor that can be set on demand.

In addition to the constraints (6), (7), and (8), there is one more constraint for LUC, which ensures that the estimated $\mathcal{L}_s(t)$ can always capture the actual $\ell_s(t)$. We have

$$\ell_s(t) \in \mathcal{L}_s(t) \text{ for } s \in \mathcal{S}. \quad (11)$$

This constraint (11) is needed to guarantee the effectiveness of incumbent protection and interference mitigation in Problem 1. If any $\ell_s(t) \notin \mathcal{L}_s(t)$ occurs, even though constraint (6) makes sure that $\mathcal{L}_s(t)$ does not overlap with any primary exclusion zone, it is likely that $\ell_s(t)$ moves into an exclusion zone, resulting in harmful interference from user s to primary users. Without constraint (11), it is also possible that the value of $I(t)$ computed using $\mathcal{L}_s(t)$ cannot accurately reflect the actual interference from or to user s located at $\ell_s(t)$.

In summary, the problem of LUC solved at the mobile users in time slot t can be written as follows.

Problem 2 (Location Update Control)
 Given: $A_s(t')$ for $t' = t, t-1, \dots, s \in \mathcal{S}$;
 $X_{s,k}(t), Y_{s,k}(t)$ for $s \in \mathcal{S}, k \in \mathcal{K}$;
 $X_{s,k}(t+1), Y_{s,k}(t+1)$ for $s \in \mathcal{S}, k \in \mathcal{K}$;
 Find: $A_s(t+1)$ for $s \in \mathcal{S}$;
 Minimize: $G(t+1)$;
 Subject to: (6), (7), (8), (11).

If Problem 2 is solved centrally at the database, constraint (11) cannot be guaranteed. Because user s does not have to continuously query and update the database in every time slot, the database cannot always keep track of $\ell_s(t)$. It is possible that a rapid change in a user's speed or direction makes the estimated movement contour become inaccurate and lose track of the user. This constraint, however, can be easily satisfied at a user. Each user s knows $\ell_s(t)$ (e.g., via GPS) and its past location update history. The user can locally compute $\mathcal{Q}_s(t)$ ($\mathcal{L}_s(t)$) using the same movement prediction model as used at the database [16] and compare it with $\ell_s(t)$ to see whether constraint (11) holds. In the rest of this section, we will derive a local heuristic solution to Problem 2 through a local strategy for setting each $A_s(t+1)$.

5.2 A Local Two-Level Strategy

Solving Problem 2 locally at mobile users is challenging. Each user s cannot have the future knowledge of $\mathcal{Q}_{s'}(t+1)$ for $s' \neq s$ and $X_{s,k}(t+1)$, $Y_{s,k}(t+1)$ for $s \in \mathcal{S}$. As a

heuristic, we can assume a short-term prediction based on the correlation between time slot t and time slot $t+1$ [17] and the fact that the local decision of a user has limited impact on the global decision for all the users. We have

$$\begin{aligned} X_{s,k}(t+1) &= X_{s,k}(t), \\ Y_{s,k}(t+1) &= Y_{s,k}(t) \text{ for } s \in \mathcal{S}, k \in \mathcal{K}. \end{aligned} \quad (12)$$

Then, each user s only needs to set its own $A_s(t+1)$ as a part of a solution to Problem 2. However, user s cannot have the global knowledge of $\mathcal{Q}_{s'}(t)$, $X_{s',k}(t)$, $Y_{s',k}(t)$ for $s' \neq s$. Keeping track of such information locally at each user demands real-time message exchanges among all the users at all time. This is prohibitively expensive and very impractical. Thus, we need to propose a local strategy for each user s to find the proper setting of $A_s(t+1)$ that solves Problem 2 only based on local measurements.

The basic idea for our heuristic local strategy is based on locally evaluating the impact of a location update on the minimization of mutual interference $I(t+1)$ in (10). Local measurements of user movement and co-channel interference can be used to set $A_s(t+1)$ through a two-level strategy. On the first level, a *movement-driven* strategy (MDS) identifies the “must-update” and “no-update” instances with regard to the constraints of Problem 2. A location update is necessary if skipping it will violate constraint (11) or harm spectrum utilization greatly due to white space loss caused by constraint (6). A location update is avoidable if skipping it will not negatively affect interference mitigation and constraint satisfaction. For the instances when MDS is not sufficient to tell the impact of a location update, on the second level, an *interference-driven* strategy (IDS) further identifies the “no-update” instances from these uncertain instances with regard to the objective function of Problem 2. A location update is unnecessary in this step if doing it will not improve interference mitigation. In the rest of this section, we will first illustrate the designs of the two strategies, and then combine them as a local heuristic algorithm for LUC.

5.2.1 Movement-Driven Strategy

The MDS for LUC locally identifies the “must-update” and “no-update” instances with regard to the constraints of Problem 2. These instances are identified according to two design needs. First, constraint (11) should always hold. Second, constraint (6) should not lead to inefficient spectrum utilization. To explain our design formally, we will show how we mathematically quantify the location accuracy and location fuzziness of $\mathcal{Q}_s(t)$ for each user $s \in \mathcal{S}$ and how to use these two values in meeting the two needs.

We define the *location accuracy* of $\mathcal{Q}_s(t)$ by $q_s(t) \triangleq q_{s,\ell_s}(t)$, which is the probability of locating $\ell_s(t)$ by $\mathcal{Q}_s(t)$. The first need is addressed by the following two propositions.

Proposition 1 For each user $s \in \mathcal{S}$, constraint (11), i.e., $\ell_s(t) \in \mathcal{L}_s(t)$, is locally guaranteed if $q_s(t) > 0$ holds.

Proof Please refer to the definition of $\mathcal{L}_s(t)$. \square

In robot navigation research, a LPG is often created by using Bayes filters, which probabilistically estimate the location of an object from noisy observations. In our problem, however, each mobile user exactly reports its location information. Hence, we only need to implement one prediction step of general Bayes filters without sequential localization error correction. We integrate grid-based localization [18] and sampling-based localization [19, 20] to generate LPGs.

Specifically, the database can utilize the sequence of $A_s(t')$ for $t' = t, t-1, \dots$ and the past location update history to generate $\mathcal{Q}_s(t)$. Suppose that the latest location update

from user s happens in time slot t_0 , i.e., $A_s(t_0) = 1$ and $A_s(t') = 0$ for $t' = t, t-1, \dots, t_0 + 1$. Define $\tilde{t}_s(t) = t - t_0$ as the elapsed time since the last location update. The past location update history can include the reported location \tilde{l}_s and velocity vector $\tilde{\mathbf{v}}_s$ in time slot t_0 . In each prediction step for locating user s using a technique of sampling-based Monte Carlo localization, all the samples for location estimation are placed at \tilde{l}_s initially. Then, each sample is shifted according to its randomly generated succeeding location

$$\dot{L}_s(t) \triangleq \tilde{l}_s + (\tilde{\mathbf{v}}_s + \tilde{\mathbf{w}}_s) \tilde{t}_s(t), \quad (13)$$

where $\dot{L}_s(t)$ is a random variable generated by the movement prediction model representing an estimate of $\ell_s(t)$, and $\tilde{\mathbf{w}}_s$ is a vector of random variables characterizing the uncertainty in generating $\mathcal{Q}_s(t)$. The $\mathcal{Q}_s(t)$ is essentially the probability distribution of $\dot{L}_s(t)$ over \mathcal{L} . Using (13), we have

$$q_s(t) = Pr\{\tilde{l}_s + (\tilde{\mathbf{v}}_s + \tilde{\mathbf{w}}_s) \tilde{t}_s(t) = \ell_s(t)\}. \quad (14)$$

The probability distribution of $\tilde{\mathbf{w}}_s$ varies with the applied movement prediction model and the knowledge of underlying terrain data. As an example, we assume $\tilde{\mathbf{w}}_s$ follows a bivariate normal distribution with parameters $(\mu_x = 0, \mu_y = 0, \sigma_x^2, \sigma_y^2, \rho)$ in the two-dimensional map grid with no terrain knowledge. Then, we have the following common results.

Proposition 2 For bivariate normal $\tilde{\mathbf{w}}_s$, $q_s(t)$ is upper bounded by a monotonically decreasing function of $\tilde{t}_s(t)$, i.e.,

$$\tilde{q}_s(t) = Pr\{\tilde{l}_s + (\tilde{\mathbf{v}}_s + \tilde{\mathbf{w}}_s) \tilde{t}_s(t) = E[\dot{L}_s(t)]\},$$

where $E[\dot{L}_s(t)] = \tilde{l}_s + \tilde{\mathbf{v}}_s \tilde{t}_s(t)$. As $\ell_s(t)$ deviates from $E[\dot{L}_s(t)]$, $q_s(t)$ is a monotonically decreasing function of $\tilde{t}_s(t)$.

Proof Let $\tilde{\mathbf{w}}_s = (W_x, W_y)$ and define a vector of random variables $(L_x, L_y) = \tilde{t}_s(W_x, W_y) + (\tilde{l}_x, \tilde{l}_y)$, where $(\tilde{l}_x, \tilde{l}_y) = \tilde{l}_s + \tilde{\mathbf{v}}_s \tilde{t}_s$ can be viewed as a constant vector. Based on the probability density function (PDF) of (W_x, W_y) and the properties for linear transformations of bivariate random variables, the joint PDF of (L_x, L_y) can be derived as

$$f_{L_x, L_y}(l_x, l_y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \frac{1}{\tilde{t}_s^2} \times \exp\left(-\frac{\left(\frac{l_x-\tilde{l}_x}{\sigma_x}\right)^2 + \left(\frac{l_y-\tilde{l}_y}{\sigma_y}\right)^2 - 2\rho\left(\frac{l_x-\tilde{l}_x}{\sigma_x}\right)\left(\frac{l_y-\tilde{l}_y}{\sigma_y}\right)}{2(1-\rho^2)} \frac{1}{\tilde{t}_s^2}\right). \quad (15)$$

Hence, $q_s(t)$ is the double integral of $f_{L_x, L_y}(l_x, l_y)$ over the small area of location pixel $\ell_s(t)$, and is a function of $\tilde{t}_s(t)$ for $\tilde{t}_s(t) > 0$. Because the size of $\ell_s(t)$ is fixed, $q_s(t)$ is only determined by $f_{L_x, L_y}(l_x, l_y)$. For $\rho \in (-1, 1)$, we know that $\frac{\left(\frac{l_x-\tilde{l}_x}{\sigma_x}\right)^2 + \left(\frac{l_y-\tilde{l}_y}{\sigma_y}\right)^2 - 2\rho\left(\frac{l_x-\tilde{l}_x}{\sigma_x}\right)\left(\frac{l_y-\tilde{l}_y}{\sigma_y}\right)}{2(1-\rho^2)} \geq 0$. Hence,

$$f_{L_x, L_y}(l_x, l_y) \leq \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \frac{1}{\tilde{t}_s^2}. \quad (16)$$

The equality in (16) holds when $(l_x, l_y) = (\tilde{l}_x, \tilde{l}_y)$. In this perfect case, $\ell_s(t)$ can always be captured by $E[\dot{L}_s(t)]$, which is on the top of the bell-shaped joint density $f_{L_x, L_y}(l_x, l_y)$. In this case, $q_s(t)$ reaches its upper bound $\tilde{q}_s(t)$, which is the double integral of $f_{L_x, L_y}(l_x, l_y)$ over $\ell_s(t) = E[\dot{L}_s(t)]$. Since $\tilde{q}_s(t)$ is determined by $f_{L_x, L_y}(l_x, l_y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \frac{1}{\tilde{t}_s^2}$, $\tilde{q}_s(t)$ is a monotonically decreasing function of $\tilde{t}_s(t)$.

As $\ell_s(t)$ deviates from $E[\dot{L}_s(t)]$, $d_{\ell_s, \tilde{l}_s}(t) < d_{\ell_s, \tilde{l}_s}(t+1)$, where $d_{\ell_s, \tilde{l}_s}(t)$ is the distance between $\ell_s(t)$ and $E[\dot{L}_s(t)]$. It is easy to show $q_s(t) > q_s(t+1)$ through (15). \square

Even though $q_s(t)$ may have local fluctuations when $\tilde{t}_s(t)$ increases due to likely irregular trajectory of user s , it is bounded by the monotonically decreasing upper bound and it is common that $\ell_s(t)$ deviates from $E[\dot{L}_s(t)]$ without localization correction. Its general trend is decreasing with time. Hence, setting $A_s(t+1) = 0$ will mostly lead to a further decrease in $q_s(t)$. To ensure $q_s(t+1) > 0$, user s needs to track the drop in $q_s(t)$ and set $A_s(t+1) = 1$ when $q_s(t)$ is too low. This addresses the first need for constraint (11).

We define the *location fuzziness* of $\mathcal{Q}_s(t)$ by $|\mathcal{L}_s(t)|$, which is the covered pixel area of $\mathcal{L}_s(t)$. The second need can be addressed by the following proposition.

Proposition 3 For bivariate normal $\tilde{\mathbf{w}}_s$, $|\mathcal{L}_s(t)|$ is a monotonically increasing function of $\tilde{t}_s(t)$.

Proof The size of $|\mathcal{L}_s(t)|$ depends on the covariance matrix of $(L_x, L_y) = \tilde{t}_s(W_x, W_y) + (\tilde{l}_x, \tilde{l}_y)$, which is

$$\begin{bmatrix} \tilde{t}_s^2 \sigma_x^2 & \rho \tilde{t}_s \sigma_x \tilde{t}_s \sigma_y \\ \rho \tilde{t}_s \sigma_x \tilde{t}_s \sigma_y & \tilde{t}_s^2 \sigma_y^2 \end{bmatrix},$$

where both $\tilde{t}_s \sigma_x$ and $\tilde{t}_s \sigma_y$ grow with $\tilde{t}_s(t)$. Hence, for a fixed ρ , $|\mathcal{L}_s(t)|$ enlarges as $\tilde{t}_s(t)$ increases. \square

Hence, setting $A_s(t+1) = 0$ will lead to a further increase in $|\mathcal{L}_s(t)|$, which indicates that $\mathcal{L}_s(t+1)$ is a fuzzier estimate of $\ell_s(t+1)$ compared with $\mathcal{L}_s(t)$ as an estimate of $\ell_s(t)$. To avoid significant loss of white space opportunities and high chance of inaccurate resource allocations, user s needs to set $A_s(t+1) = 1$ when necessary. This addresses the second need for not being affected by constraint (6) too much.

If $\tilde{\mathbf{w}}_s$ follows a general probability distribution, different movement prediction models can achieve different levels of location estimation deviation, but the properties of lower location accuracy and greater location fuzziness are usually true as the elapsed time increases.

Local MDS The MDS works as follows. Each user s can track $q_s(t)$ and set $A_s(t+1)$ using two thresholds: α_1 and α_2 ($\alpha_1 \geq \alpha_2$). There are three possible cases:

- When $q_s(t) \geq \alpha_1$, user s assumes “no-update” case and sets $A_s(t+1) = 0$.
- When $\alpha_2 < q_s(t) < \alpha_1$, user s assumes uncertain case and needs to consider using IDS for further investigation.
- When $q_s(t) \leq \alpha_2$, user s assumes “must-update” case and sets $A_s(t+1) = 1$.

For case a, a relatively small $|\mathcal{L}_s(t)|$ ensures that $\mathcal{L}_s(t)$ nicely captures $\ell_s(t)$ and thus constraints (11) and (6) are easy to satisfy. If user s sets $A_s(t+1) = 1$, it is very likely that this action has little impact on the resource allocation decision under the slightly reduced location uncertainty. For case c, a relatively large $|\mathcal{L}_s(t)|$ tends to cause significant loss of white space opportunities due to constraint (6) and high chance of inaccurate interference estimation. If user s sets $A_s(t+1) = 0$, it is very likely that the database makes a suboptimal resource allocation decision under the increased location uncertainty. More importantly, if $q_s(t)$ is very low, $q_s(t+1) = 0$ is possible and thus violates constraint (11). For case b, IDS is needed as an additional step to set $A_s(t+1)$.

5.2.2 Interference-Driven Strategy

The IDS for LUC further locally identifies the “no-update” instances among the uncertain instances from MDS (in case b) with regard to the objective function of Problem 2. To decide whether $A_s(t+1)$ should be 0 or 1, user s needs to predict which action will lead to a smaller objective value. Ideally, user s can take the following process. First, user s solves Problem 1 for time slot $t+1$ to obtain the values of

$I(t+1)$ for both $A_s(t+1) = 0$ and $A_s(t+1) = 1$ cases. Next, user s compares the values of $G(t+1)$ for the two cases. Whichever objective value that comes the smallest corresponds to the optimal setting of $A_s(t+1)$ for Problem 2. However, this process requires user s to have perfect global knowledge, which is impractical to obtain. Thus, we need to derive a heuristic strategy to set $A_s(t+1)$, which can approximate this process but only requires local knowledge.

Based on (9) and (10), we know that the key in the design of IDS is to find a local way to determine whether

$$I(t+1)|_{A_s(t+1)=0} - I(t+1)|_{A_s(t+1)=1} > \frac{(1-\delta)\theta}{\delta|S|}. \quad (17)$$

If (17) is likely to be true (false), set $A_s(t+1) = 1$ (0).

Global IDS To find a way to judge (17), we consider that user s using a channel $k \in \mathcal{K}$ has a set of co-channel neighbors $s' \in \mathcal{S}_k(t)$, $s' \neq s$. As an example, we discuss IDS with global knowledge based on the co-channel interference in the uplink from user s to others s' . Define $d_{s'}(t)$ as the protection radius from the receiver of each interfered link s' , and define $d_{s,s'}(t)$ as the distance between the transmitter of the interfering link s and the receiver of each interfered link s' . Define $\omega_1 \in [0, 1)$ and $\omega_2 \in (1, \infty)$. With proper ω_1 and ω_2 , there are three possible relative location cases:

1. When $d_{s,s'}(t) \leq \omega_1 d_{s'}(t)$ for any $s' \in \mathcal{S}_k(t)$, $s' \neq s$, user s assumes “no-update” case and sets $A_s(t+1) = 0$.
2. When $\omega_1 d_{s'}(t) < d_{s,s'}(t) < \omega_2 d_{s'}(t)$ for any $s' \in \mathcal{S}_k(t)$, $s' \neq s$, user s sets $A_s(t+1) = 1$.
3. When $d_{s,s'}(t) \geq \omega_2 d_{s'}(t)$ for all $s' \in \mathcal{S}_k(t)$, $s' \neq s$, user s assumes “no-update” case and sets $A_s(t+1) = 0$.

For case 1, user s is deep inside the protection area of a link s' . Because MDS guarantees that $\ell_s(t) \in \mathcal{L}_s(t)$ and $|\mathcal{L}_s(t)|$ is not too large in case b, there should be a relatively full overlap between $\mathcal{L}_s(t)$ and the protection area of link s' . Thus, even setting $A_s(t+1) = 0$, Problem 1 is likely to judge that links s and s' have a high probability of mutual interference and make a resource allocation decision as same as that in the case of setting $A_s(t+1) = 1$ (so that the mutual interference probability is 1). Because the decisions for WSA in both update and no-update cases should be the same, (17) is likely to be false. For case 3, user s is far outside the protection area of any link $s' \in \mathcal{S}_k(t)$. The movement contour of user s is likely to be mostly outside the protection area of any link s' . Thus, even setting $A_s(t+1) = 0$, Problem 1 is likely to judge that links s and s' have a low probability of mutual interference and make a resource allocation decision as same as that in the case of setting $A_s(t+1) = 1$ (so that the mutual interference probability is 0). Because the decisions for WSA in both update and no-update cases should be the same, (17) is likely to be false. For case 2, user s is near the boundary of the protection area of a link s' . The movement contour of user s is partly inside and partly outside the protection area of link s' . If user s sets $A_s(t+1) = 1$, Problem 1 can precisely know whether there is mutual interference between links s and s' . If there exists mutual interference, the database will try to reallocate channels to links s and s' to eliminate the interference. If there exists no mutual interference, the database is likely to keep the interference to be zero. In both cases, the solution to Problem 1 usually makes the probability of mutual interference between links s and s' be zero. If user s sets $A_s(t+1) = 0$, Problem 1 can only use the movement contour to obtain a positive value of mutual interference probability. The left-hand side of (17) is usually large, so (17) is likely to be true.

Unfortunately, user s cannot precisely know which case it belongs to without perfect knowledge of other co-channel

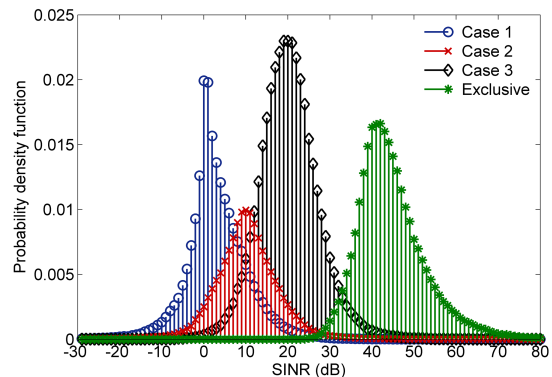


Figure 2: PDFs of $R_{s,k}(t)$ ($\omega_1 = \frac{2}{3}$, $\omega_2 = \frac{4}{3}$).

users’ location information, which is impractical to obtain. However, user s can estimate which relative location case it is likely in based on its local measurement of $R_{s,k}(t)$, the SINR that user s has observed on channel k in time slot t .

Proposition 4 The relative location cases and the local measurement of $R_{s,k}(t)$ are correlated in a stochastic sense.

For example, we use numerical results to create Figure 2, which shows the PDFs of $R_{s,k}(t)$ observed in different relative location cases. The exclusive case refers to the situation where there is no any other user sharing the channel with user s . Obviously, a location update is not needed in the exclusive case. The PDFs of the other three cases are averaged when the number of co-channel users ranges from 2 to 7. The details of the 7-cell system will be presented in section 6. We observe that when the locally measured $R_{s,k}(t)$ is either low enough or high enough, the false judgement rates for the “no-update” cases (except for case 2) are relatively low due to their distinct peaks in the PDFs. Then, user s can usually identify the “no-update” instances properly.

Local IDS The IDS works as follows. Each user s that takes a channel k can measure $R_{s,k}(t)$ and set $A_s(t+1)$ using two thresholds: β_1 and β_2 ($\beta_1 \leq \beta_2$). There are three possible cases mapping with the relative location cases:

- 1’ When $R_{s,k}(t) \leq \beta_1$, user s assumes “no-update” case and sets $A_s(t+1) = 0$.
- 2’ When $\beta_1 < R_{s,k}(t) < \beta_2$, user s sets $A_s(t+1) = 1$.
- 3’ When $R_{s,k}(t) \geq \beta_2$, user s assumes “no-update” case and sets $A_s(t+1) = 0$.

For case 1’, user s is most likely in case 1. For case 3’, user s is very likely in either case 3 or the exclusive case. For case 2’, user s can be in case 1, 2, or 3. Using a safe strategy that cares more about interference mitigation effectiveness than database query cost, user s assumes the worst case 2.

5.2.3 Combining MDS and IDS

The local two-level strategy for LUC that combines MDS and IDS is summarized in Algorithm 2. Two sets of thresholds, i.e., movement-related α_1 and α_2 , and interference-related β_1 and β_2 , are used to identify the cases where a location update is unnecessary (in cases a, 1’, 3’) or necessary (in cases c, 2’). Running Algorithm 2 at mobile users gives a heuristic solution to Problem 2. After the thresholds have been determined, the complexity of Algorithm 2 mainly comes from computing $\mathcal{Q}_s(t)$ through one prediction step of a Bayes filter, which is not very expensive. In sections 6 and 7, we will discuss how these thresholds can be set.

6. PERFORMANCE EVALUATION

In this section, we evaluate the proposed algorithms using the results from two simulation experiments. First, we have

Algorithm 2 for LUC at user $s \in \mathcal{S}$ in time slot t

```

1: if  $A_s(t) == 1$  (0) then
2:   query (do not query) the database
3: end if
4: access white space spectrum according to  $X_{s,k}(t)$ ,
    $Y_{s,k}(t)$  for  $k \in \mathcal{K}$  assigned by the database
5: compute  $Q_s(t)$  as in Algorithm 1, and measure  $q_s(t)$ 
6: if case a (c) then
7:   set  $A_s(t+1) = 0$  (1)
8: else
9:   set  $A_s(t+1) = 0$ 
10:  measure  $R_{s,k}(t)$  for all  $k$  with  $X_{s,k}(t) == 1$ 
11:  if case 2' for any  $k$  then
12:    set  $A_s(t+1) = 1$ 
13:  end if
14: end if

```

simulated WiFi-like secondary networks using a theoretical mobility model. Second, we have simulated cellular-like secondary networks using a real-world mobility trace dataset.

6.1 Model-Driven Simulation

In the first experiment, we study a 7-cell spectrum sharing system, in which the seven base stations for secondary networks are placed at the center and six vertices of a regular hexagon with the edge length of 1500 m. Total m channels are shared by m primary users and $2m$ secondary users. Each primary user occupies a different channel in its exclusion zone with the radius of 1500 m, and its duty cycle is λ . The primary users are uniformly distributed. Each secondary user is randomly associated with a base station, and let $\bar{K}_s = 1$, $\bar{Y}_s = 100$ mW, $\bar{R}_s = 10$ dB. The random movement of each user is defined by a semi-Markov smooth mobility model [21]. The target direction in the model is restricted back when the user is more than 1500 m away from its home base station, and the average target speed is on the order of human walking speed. The granularity of the map grid is $50\text{m} \times 50\text{m}$, and the duration of a time slot is 120 s. In this example, the cells largely overlap with each other, and user mobility relative to the size of map grid is high enough to generate dynamic inter-cell interference. We explain the settings of α_1 , α_2 in MDS and β_1 , β_2 in IDS to adapt the database query intervals. The average objective function is recorded as $\bar{G} = \delta \bar{I} + (1 - \delta) \bar{C}$, where \bar{I} is the average fraction of time for unsuccessful reception in each link and \bar{C} is the average value of $C(t)$ over time with $\theta = 1$.

First, we show the impact of α_1 and α_2 on \bar{G} . In Figure 3, fixing $\alpha_2 = 0.01$, $\beta_1 = 5$ dB, $\beta_2 = 14$ dB, we increase α_1 from 0 to 1. This increases the probability of case 2' in IDS and thus increases \bar{C} . In Figure 4, fixing $\alpha_1 = \alpha_2$, we increase α_2 from 0 to 1. This increases the probability of case c in MDS and thus increases \bar{C} . To balance between \bar{I} and \bar{C} , larger α_1 and larger α_2 are chosen for larger δ to keep higher location accuracy and smaller location fuzziness. Larger $\alpha_1 - \alpha_2$ is used to give more preference to IDS.

Second, we show the impact of β_1 and β_2 on \bar{G} . According to Figure 2, too small β_1 or too large β_2 can create a lot of unnecessary location updates due to falsely treating case 1 or 3 as case 2'. On the contrary, too large β_1 or too small β_2 can omit a lot of necessary location updates due to falsely treating case 2 as case 1' or 3'. The values of β_1 and β_2 should be chosen with regard to the false judgement rates for the relative location cases. In Figure 5, fixing $\alpha_1 = 0.7$, $\alpha_2 = 0.01$, we increase $\beta_2 - \beta_1$ from 0 (decreasing β_1 from 9.5 dB and increasing β_2 from 9.5 dB). This increases

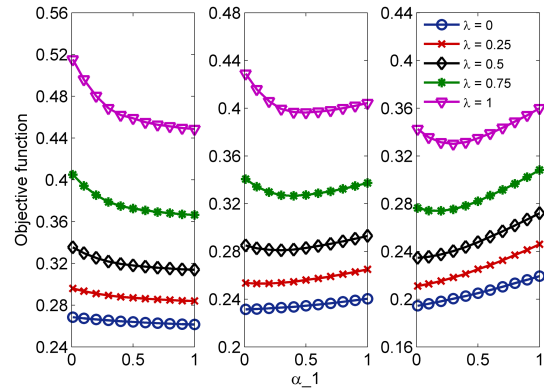


Figure 3: Impact of α_1 on \bar{G} ($m = 28$):
(a) $\delta = 1$ (left); (b) $\delta = 0.8$ (middle); (c) $\delta = 0.6$ (right).

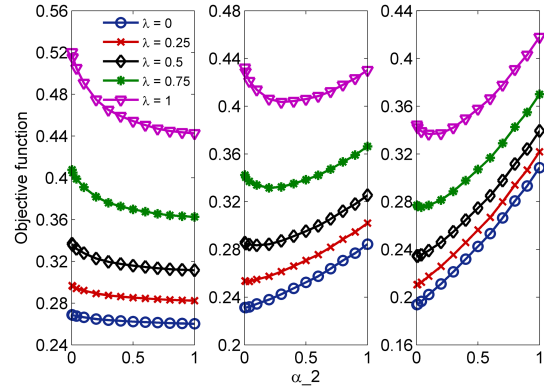


Figure 4: Impact of α_2 on \bar{G} ($m = 28$):
(a) $\delta = 1$ (left); (b) $\delta = 0.8$ (middle); (c) $\delta = 0.6$ (right).

the probability of case 2' in IDS and thus increases \bar{C} . To balance between \bar{I} and \bar{C} , smaller β_1 and larger β_2 are chosen for larger δ to reduce the false judgements that cause missing necessary location updates, while larger β_1 and smaller β_2 are chosen for smaller δ to reduce the false judgements that cause unnecessary location updates.

Third, we compare the value of $1 - \bar{I}$ given by our heuristic mechanism with the value given by an impractical optimal mechanism. Here, our mechanism is called movement-driven and interference-driven database query strategy (MIDQ). We also consider two alternative heuristic mechanisms: a pure movement-driven database query strategy (MDQ) (by fixing $\alpha_1 = \alpha_2$) and a periodic database query strategy (PDQ) (as used in [4]). The optimal mechanism assumes perfect global knowledge, so that all the users' actual locations are known (without location update cost) and an optimal resource allocation is performed accordingly. The ratio of $1 - \bar{I}$ from a heuristic solution to that from the optimal solution is depicted in Figure 6. It can be seen that the ratio grows with increasing database query frequency. For MIDQ, \bar{C} is increased by first increasing α_1 ($\alpha_2 = 0.01$, $\beta_1 = 5$ dB, $\beta_2 = 14$ dB) from 0.01 to 1 and then increasing $\beta_2 - \beta_1$. For MDQ, \bar{C} is increased by increasing $\alpha_1 = \alpha_2$ from 0.01 to 1. Compared with MDQ and PDQ, the proposed MIDQ achieves better reduction of mutual interference with the same level of database query frequency (cost). With a high enough database query frequency, MIDQ can achieve almost the same interference level as the optimal mechanism.

6.2 Trace-Driven Simulation

In the second experiment, we have utilized a real-world mobility trace dataset that contains the GPS coordinates of

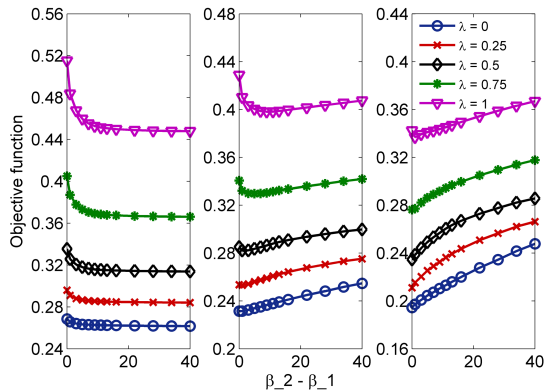


Figure 5: Impact of β_1 and β_2 on \bar{G} ($m = 28$): (a) $\delta = 1$ (left); (b) $\delta = 0.8$ (middle); (c) $\delta = 0.6$ (right).

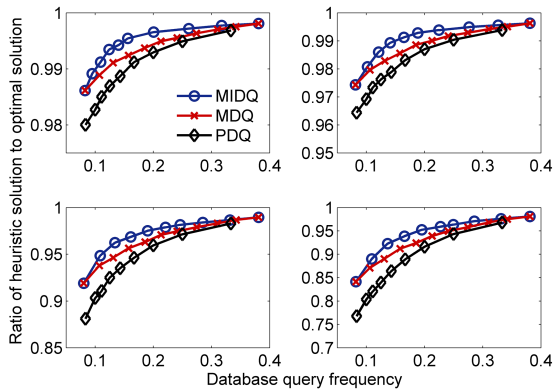


Figure 6: Optimality gap of heuristic WSA and LUC ($m = 28$): (a) $\lambda = 0$ (top-left); (b) $\lambda = 0.25$ (top-right); (c) $\lambda = 0.75$ (bottom-left); (d) $\lambda = 1$ (bottom-right).

real taxis collected in the San Francisco Bay Area [22]. In this experiment, a taxi represents a mobile secondary user. However, we need to deploy the base stations for secondary networks, since the dataset does not include any location information of cellular base stations or WiFi access points. We use a similar method as used in [23]. The possible locations of the base stations to be deployed are the location points that have been traveled by the taxis in the dataset, and the points are along roads and around buildings. Each point is assigned with a selection probability, which is the frequency at which the location point has been visited by the taxis. In this way, more base stations are deployed in more heavily traveled areas. Total m channels are shared by m primary users and $2m$ secondary users. As shown in Figure 7, the traces of 48 taxis are selected, and 12 base stations are deployed. We assume that each taxi is associated with the nearest base station, i.e., inter-cell handover is enabled. The exclusion zones of the primary users with a radius of 3000 m are uniformly distributed in the simulation area. The granularity of the map grid is $100\text{m} \times 100\text{m}$, and the duration of a time slot is 60 s. Similar to Figure 6, the ratio of the heuristic solution to the optimal solution is shown in Figure 8. From the results, we can see that the proposed MIDQ still outperforms MDQ and PDQ. The overall performance of the heuristic mechanisms degrades in this experiment compared with that in the first experiment. Besides the impact of incumbents, this should be caused by the high-speed vehicular mobility that leads to less accurate movement prediction and the dynamic user-cell association that leads to less accurate local interference estimation.

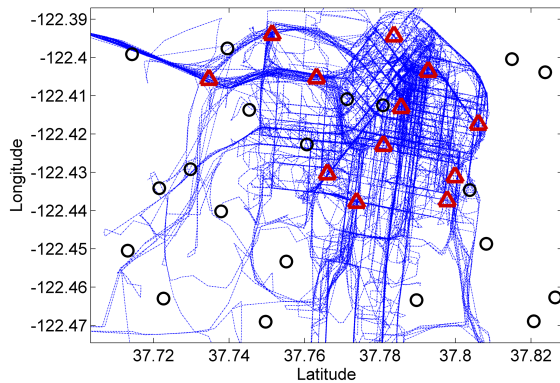


Figure 7: A trace-driven simulation scenario in San Francisco (lines: taxi movement trajectories; triangles: secondary base stations; circles: primary exclusion zone centres).

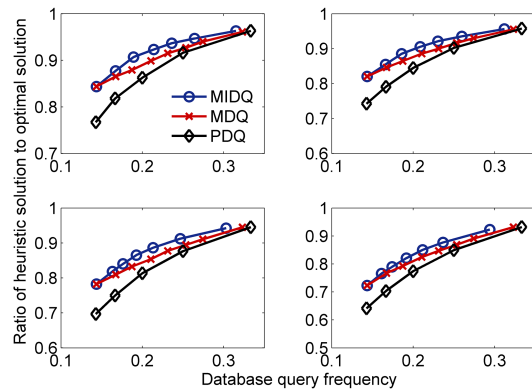


Figure 8: Optimality gap of heuristic WSA and LUC (traces): (a) $\lambda = 0$ (top-left); (b) $\lambda = 0.25$ (top-right); (c) $\lambda = 0.75$ (bottom-left); (d) $\lambda = 1$ (bottom-right).

7. DISCUSSIONS

In this section, we discuss system design guidelines and real-world deployment issues in database-driven OSA.

1 The database query interval of each mobile user should be adapted according to both internal factors (e.g., user mobility, database processing capacity, and location privacy) and external factors (e.g., incumbent protection, and network coexistence) to achieve a proper balance between spectrum sharing and database access. In general, a higher frequency of database queries is necessary if i) user movement pattern is more dynamic; ii) interference mitigation effectiveness is of higher priority than database query cost; iii) location information is less sensitive at the database; iv) the duty cycle of primary users is larger, i.e., less channels are available for certain users; v) mutual interference is more likely to occur, i.e., more users share certain channels.

2 In a probabilistic coexistence framework as proposed in this work, the database can set the system parameters offline (e.g., via statistical analysis) or online (e.g., via a stochastic decision process). For example, the thresholds α_1 and α_2 can be empirically set according to the maximum and minimum desired levels of location accuracy or location fuzziness, respectively, for a certain movement prediction model. The thresholds β_1 and β_2 can be dynamically adjusted according to the changes in the spectrum environment (e.g., primary user activity, secondary user density and mobility, and resource reallocation) and the results from past decisions (e.g., the rate at which erroneous decisions have been made in determining the appropriate case for a user). To reduce complexity, the thresholds can also be empirically

set according to the PDFs as in Figure 2. One can learn the amount of false judgement for each case given a particular setting of β_1 and β_2 , and thus make an educated choice. The offline and online decisions can be combined to mediate a trade-off between performance and complexity.

3 For the estimation of user movement pattern at the database, the applied localization method or movement prediction model determines how the location uncertainty level of a user changes with time. In general, a more advanced localization technique (e.g., a method that integrates GPS, maps, and user behaviors) achieves a slower drop in location accuracy and a slower rise in location fuzziness, and requires less location updates for localization error correction.

4 For the estimation of harmful interference to primary users and mutual interference among secondary users at the database, the applied radio wave propagation model affects the effectiveness of spectrum availability prediction and resource allocation. In general, a more sophisticated signal propagation model (e.g., Longley-Rice with terrain data [4]) achieves better incumbent protection and higher spectrum utilization, and even further improves local decisions at the users for mutual interference mitigation.

5 It is hard to quantify the cost of database queries in database-driven spectrum sharing. However, one of major bottlenecks in such a system is the capacity of the central database server. In a fine-grained system, each single user performs LUC, and the central server manages spectrum sharing information and performs WSA at the user-level granularity. The computational complexity of resource allocation is affected by the number of registered users and the spatial granularity of map grid, and the resource allocation has to meet the timely requirement if a time slot is very short. Hence, when designing a database-driven spectrum sharing system, the maximum number of registered users that can be served by each server needs to be considered. Alternatively, each base station can manage the users in its cell as a whole. In such a coarse-grained system, each base station performs LUC, and the central server offers the services at the cell-level granularity. In this approach, each base station further performs WSA at the user-level granularity. In general, the fine-grained system is recommended when database processing capacity is not a problem, while the coarse-grained system is preferred when server response delay or resource reallocation frequency is too high.

8. CONCLUSION

In this paper, we have addressed two major problems for supporting mobile users in database-driven OSA: WSA and LUC. In the proposed coexistence framework, WSA is performed centrally, while LUC is performed locally. We have used a LPG to quantify the location uncertainty level of each user. The problems of WSA and LUC have been jointly solved through dynamic control of LPGs. The local strategy for adapting database query intervals combines practical MDS and IDS to reduce database query cost without negatively impacting interference mitigation effectiveness. Our model-driven and trace-driven simulation results have shown that the intervals can be adapted to yield near-optimal interference mitigation with minimal location updates.

References

[1] S. Haykin. Cognitive radio: Brain-empowered wireless communications. *IEEE J. Sel. Area. Comm.*, 23(2), Feb. 2005.
 [2] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty. Next generation/dynamic spectrum access/cognitive ra-

dio wireless networks: A survey. *Elsevier Comp. Netw.*, 50(13), Sep. 2006.
 [3] D. Gurney, G. Buchwald, L. Ecklund, S. Kuffner, and J. Grosspietsch. Geo-location database techniques for incumbent protection in the tv white space. In *Proc. IEEE DySPAN'08*, Oct. 2008.
 [4] R. Murty, R. Chandra, T. Moscibroda, and P. Bahl. Senseless: A database-driven white spaces network. *IEEE Trans. Mobi. Comp.*, 11(2), Feb. 2012.
 [5] M. Barrie, S. Delaere, G. Sukareviciene, J. Gesquiere, and I. Moerman. Geolocation database beyond tv white spaces? matching applications with database requirements. In *Proc. IEEE DySPAN'12*, Oct. 2012.
 [6] FCC. Enabling innovative small cell use in 3.5 ghz band nprm & order. GN Docket No. 12-354, FCC 12-148, Dec. 2012.
 [7] FCC. Wireless telecommunications bureau and office of engineering and technology call for technical papers on the proposed spectrum access system for the 3.5 ghz band. GN Docket No. 12-354, DA 13-2213, Nov. 2013.
 [8] C. Ghosh, S. Roy, and D. Cavalcanti. Coexistence challenges for heterogeneous cognitive wireless networks in tv white spaces. *IEEE Wirel. Comm.*, 18(4), Aug. 2011.
 [9] B. Gao, J.-M. Park, Y. Yang, and S. Roy. A taxonomy of coexistence mechanisms for heterogeneous cognitive radio networks operating in tv white spaces. *IEEE Wirel. Comm.*, 19(4), Aug. 2012.
 [10] A. W. Min, K.-H. Kim, J. P. Singh, and K. G. Shin. Opportunistic spectrum access for mobile cognitive radios. In *Proc. IEEE INFOCOM'11*, Apr. 2011.
 [11] H. Kim and K. G. Shin. Asymmetry-aware real-time distributed joint resource allocation in ieee 802.22 wrans. In *Proc. IEEE INFOCOM'10*, Mar. 2010.
 [12] Q. Ni and C. C. Zarakovitis. Nash bargaining game theoretic scheduling for joint channel and power allocation in cognitive radio systems. *IEEE J. Sel. Area. Comm.*, 30(1), Jan. 2012.
 [13] B. Gao, J.-M. Park, and Y. Yang. Uplink soft frequency reuse for self-coexistence of cognitive radio networks operating in white-space spectrum. In *Proc. IEEE INFOCOM'12*, Mar. 2012.
 [14] Y. Tachwali, B. F. Lo, I. F. Akyildiz, and R. Agustí. Multiuser resource allocation optimization using bandwidth-power product in cognitive radio networks. *IEEE J. Sel. Area. Comm.*, 31(3), Mar. 2013.
 [15] B. Gao, Y. Yang, and J.-M. Park. A credit-token-based spectrum etiquette framework for coexistence of heterogeneous cognitive radio networks. In *Proc. IEEE INFOCOM'14*, Apr. 2014.
 [16] Y. Xu, J. Winter, and W.-C. Lee. Dual prediction-based reporting for object tracking sensor networks. In *Proc. ACM MobiQuitous'04*, Aug. 2004.
 [17] Q. Xu, S. Mehrotra, Z. M. Mao, and J. Li. Proteus: Network performance forecast for real-time, interactive mobile applications. In *Proc. ACM MobiSys'13*, Jun. 2013.
 [18] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proc. AAAI'96*, Aug. 1996.
 [19] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proc. IEEE ICRA '99*, May 1999.
 [20] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. Sign. Proc.*, 50(2), Feb. 2002.
 [21] M. Zhao and W. Wang. A novel semi-markov smooth mobility model for mobile ad hoc networks. In *Proc. IEEE GLOBECOM'06*, Nov. 2006.
 [22] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. CRAWDAD data set epfl/mobility (v. 2009-02-24). online: <http://crawdad.org/epfl/mobility/>, Feb. 2009.
 [23] S. Dimatteo, P. Hui, B. Han, and V. O.K. Li. Cellular traffic offloading through wifi networks. In *Proc. IEEE MASS'11*, Oct. 2011.