This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3032434, IEEE Transactions on Mobile Computing

Spectrum Activity Surveillance: Modeling and Analysis from Perspectives of Surveillance Coverage and Culprit Detection

Jie Wang*, Wenye Wang*, Cliff Wang†, and Min Song‡

Abstract—Spectrum activity surveillance (SAS) is essential to dynamic spectrum access (DSA)-enabled systems with a two-fold impact: it is a primitive mechanism to collect usage data for spectrum efficiency improvement; it is also a prime widget to collect misuse forensics of unauthorized or malicious users. While realizing SAS for DSA-enabled systems appears to be intuitive and trivial, it is, however, a challenging yet open problem. On one hand, a large-scale SAS function is costly to implement in practice; on the other hand, it is not clear how to characterize the efficacy and performance of monitor deployment strategies. To address such challenges, we introduce a three-factor space, composed of *spectrum*, *time*, and *geographic region*, over which the SAS problem is formulated by a two-step solution: 3D-tessellation for sweep (monitoring) *coverage* and graph walk for detecting *spectrum culprits*, that is, devices responsible for unauthorized spectrum occupancy. In particular, our system model transforms SAS from a globally collective activity to localized actions, and strategy objectives from qualitative attributes to quantitative measures. With this model, we design low-cost deterministic strategies for dedicated monitors, which outperform strategies found by genetic algorithms, and performance-guaranteed random strategies for crowd-source monitors, which can detect adversarial spectrum culprits in bounded time.

1 Introduction

Dynamic spectrum access (DSA) has been envisioned as one of the key technologies for high-speed wireless systems [1], *e.g.*, 5G networks [2], since it is expected to boost spectrum efficiency by allowing wireless devices to temporally operate beyond their designated spectrum bands, so as to mitigate the gap between the increasing frequency demand and the diminishing available spectrum. DSA is important on both individual and system levels: it is essential to advanced cognitive radio (CR) technologies, *e.g.*, CR non-orthogonal multiple access (CR-NOMA) [3]; and it is also preliminary to abstraction of wireless resources in a system, *e.g.*, wireless network virtualization (WNV) [4], [5]. So multiple DSA alliances are formed to advocate, develop and standardize DSA technologies. However, despite its great potential, the open and opportunistic nature of DSA-enabled systems bears an intrinsic demand for *spectrum activity surveillance* (SAS), as both a prerequisite and a supplement to such spectrum-agile systems.

A SAS process is expected to carry out continuous scans of spectrum activities on the frequencies of interest, for the purpose of usage data collection, and spectrum regulation policing/enforcing. On a systematic level, surveillance logs reflect the spectrum usage in wireless systems, and can be analyzed for system management, as well as data disclosure [6] purposes; on an individual level, real-time spectrum usage near an individual can serve as a crude input to its spectrum sensing action [7] for opportunistic spectrum access, which is the key to DSA-enabled systems. In this regard, Google [8] and Microsoft [9] have launched their *spectrum database* projects, providing availability of TV white space over the entire United States, as a preliminary step toward the construction of *radio environment map* (REM) [10].

On the other hand, as an immediate beneficiary of the opportunistic environment toward higher spectrum efficiency, spectrum *culprits*, which refers to overly-aggressive or malicious users, may undermine the 'right-of-way' of legitimate users, and even downgrading performance of the entire system, by occupying unauthorized frequency bands that are promised to other legitimate users. This problem is especially severe in DSA-enbaled systems with distributed spectrum sharing schemes, where a simple Listen-Before-Talk (LBT) mechanism [11] is preferred due to its scalability and comparable throughput performances. In such systems, it is easy for 'smart' spectrum culprits to abuse the DSA-enabled system, owing to the application of machine learning in cognitive radios [12], [13]. Consequently, SAS is expected to act as the 'spectrum-police', detecting spectrum misuse, guarding the rights of legitimate users, and preserving forensics for further actions.

Therefore, SAS is both a premise to leverage spectrum efficiency in compliance to policy enforcement, and a proactive approach to catch the spectrum culprits. Such a system-level function of a DSA-enabled system is completed by *spectrum monitors*, who take advantage of spectrum sensing, networking, and data processing techniques, to collect occupancy measurements, and identify spectrum culprits based on collected data. In other words, a spectrum monitor is logically composed of three building blocks: sensing hardware, measurement/detection algorithm, and commu-

- *J. Wang and W. Wang are with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh NC, 27606 (e-mail: {jwang50, wwang}@ncsu.edu).*
- *C. Wang is with the Army Research Office, Research Triangle Park, NC 27709 (e-mail: cliff.wang@us.army.mil).*
- *M. Song is with the Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 (e-mail: Min.Song@stevens.edu).*

This work is partially supported by NSF CNS-1527696 and ARO W911NF-15-2-0102.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3032434, IEEE Transactions on Mobile Computing

nication protocol. For a large-scale commercial DSA-enabled system, *e.g.*, a multi-operator LTE-WiFi overlay network in the 5 GHz unlicensed frequency bands described in [11], it is necessary to include, and coordinate multiple monitors to provide reliable and timely-updated SAS results.

In this regard, existing literature on SAS can be broadly summarized into two categories: single-monitor technique, and multiple-monitor orchestration. The former develops prototypes [7], [14], techniques, and algorithms [15], [16], for individual spectrum monitors, that can effectively differentiate spectrum misuse or abnormalities from normal activities, *e.g.*, statistical significance testing [15], and the spectrum permit mechanism [16], for security enhancement and attack mitigation. In contrast, the latter focuses on deployment and cooperation of *multiple* monitors for the purpose of better surveillance coverage [17], lower switching cost [18], or faster detection of culprits [19]. To this end, spectrum occupancy measurement with dedicated monitors has been studied in [10], while the crowd-source sensing/monitoring paradigm is proposed for cost reduction, taking advantage of collaboration [18] and distributed data decoding [20].

In prior studies of multiple monitor deployment strategies (*e.g.*, [17], [18], [19]), an implicit assumption is made for spectrum monitors to be sufficiently powerful, such that they can watch over the entire geographical region of interest and tune/move without any limit. The fact, however, is that most spectrum activities, including communications, attacks/jamming and monitoring/sniffing, are *local*, *i.e.*, confined in both the frequency domain and the space domain during a fixed-length time interval, as noted in prototype design [14], and spectrum occupancy measurements [10]. This discrepancy is especially pronounced in wideband wide-area monitoring, *e.g.*, spectrum database or REM construction, which naturally leads to an open question: *how to perform spectrum activity surveillance (SAS) and design SAS strategies (with multiple monitors) for DSA-enabled systems*?

Hindered by the constraints on spectrum license and high deployment expenses, studying the SAS problem via field tests is not a viable option, especially at the early stage when development of prototypes [7], [14], as well as standardization for CR and DSA, are still underway. Therefore, considering various monitor settings and SAS scenarios, this paper takes a modeling approach to study SAS processes from perspectives of surveillance coverage and culprit detection. Seemingly trivial, the SAS problem is actually challenging due to the following reasons. First, objectives of SAS, such as data collection and culprits detection, are by-and-large global and collective, lacking a consolidated measure, through which a SAS strategy can be fairly evaluated. Second, if spectrum is considered as a 1-D domain, the surveillance problem over a geographical region is naturally extended to a 3-D space, in which tracking surveillance coverage is non-trivial.

To address these challenges, we construct a *spectra-location space* that incorporates spectra, temporal and geographical domains, in which the locality of spectrum activities are captured by limited range and closed spaces. With respect to the modeling, design, and analysis of a SAS process, our contributions can be summarized as follows:

Modeling: We formally define *monitoring power*, *switching cost*, and *switching capacity* to characterize monitors' and culprits' activities, and formulate the SAS process into a tractable graph walk process with space-tessellation, such that a collective surveillance function are transformed into localized (even distributed) actions of individual monitors.

Metrics: We translate the qualitative data collection and culprit detection objectives of SAS processes into two quantitative metrics in the time domain, *i.e.*, the *coverage time* and *detection time*, such that different SAS (monitors deployment) strategies can be evaluated, and fairly compared.

Strategy Design: We present a deterministic SAS strategy with low switching cost for systems with dedicated spectrum monitors, and randomized strategies specialized to protect against adversarial spectrum culprits, which is suitable for crowd-source surveillance scenarios. Despite the switching capacity limit, randomized strategies of m monitors can achieve a full sweep coverage over a spectra-location space of n assignment points in $\Theta(\frac{n}{m}\ln n)$ time, and detect a persistent or adversarial culprit in $\Theta(\frac{n}{m})$ time.

This paper focuses on modeling and analysis of the *efficacy* of SAS strategies from perspectives of *coverage* and *detection*. The rest of this paper is organized as follows. We describe the system model, define performance metrics, and formulate the SAS problem in Sec. 2. Then a two-step solution is proposed in Sec. 3 to make the problem tractable. Sec. 4 presents a deterministic strategy to achieve low switching cost. Addressing the detection of adversarial culprits, randomized monitoring strategies without and with switching capacity limit are proposed and examined in Sec. 5 and Sec. 6, respectively. Sec. 7 concludes this paper.

2 Problem Formulation

In this section, we formally define the spectra-location space[1], spectrum activities and performance metrics to formulate the spectrum activities surveillance (SAS) problem.

2.1 Preliminaries

Let time t proceed in discrete steps, *i.e.*, $t \in \mathcal{T} = \{1, 2, \cdots\}$. Consider a DSA-enabled system that is deployed in a geographical region $\mathcal{A} \subset \mathbb{R}^2$. The spectrum of interest, $\mathcal{S}$, refers to the spectrum blocks that are shared[2] among K radio access technologies $\{\text{RAT}_i\}_{i=1}^K$ allowed in this system.

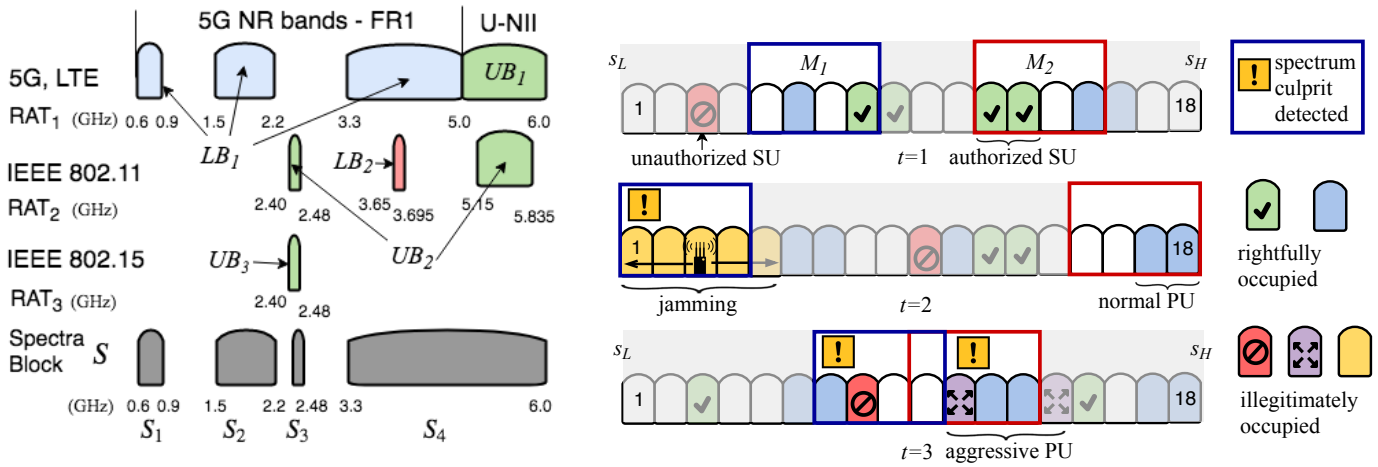
2.1.1 Spectra of Interest $\mathcal{S}$

Each RAT_i has a licensed band LB_i exclusively reserved for authorized RAT_i users, and an unlicensed band UB_i to be shared with users accessing via other RAT's. Each LB_i or UB_i can be viewed as an *interval* identified by the lowest and highest frequency as its endpoints (or a union of such intervals), then the union of all licensed and unlicensed bands, $\mathcal{S} := \cup_{i=1}^K \{\text{LB}_i \cup \text{UB}_i\}$, is the target of a SAS process in a DSA-enabled system.

1. The notion of spectra-location space is first introduced in our prior work [21], [22]. In this paper, this concept is re-defined with multiple real system settings, *e.g.*, LTE, 5G, WiFi, *etc.* taken into consideration.

2. There are two spectrum-sharing scopes for a DSA-enabled system: the inter-technology DSA, which only shares the unlicensed spectrum bands, *e.g.*, [23], and the *spectrum commons*, in which licensed bands are also included and each device has equal spectrum access right on a cost basis, *e.g.*, [11]. Both scopes can be described by our model.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3032434, IEEE Transactions on Mobile Computing



(a) An Example of the Spectra Block $\mathcal{S}$ (b) Spectrum Activities on a Continuous $\mathcal{S}$

Fig. 1. Consider a DSA-enabled system that allows access via 5G/LTE, WiFi, and Bluetooth on the sub-6GHz frequency bands. The entire spectra block is a union of disjoint continuous spectrum trunks, that is, $\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3 \cup \mathcal{S}_4$. Now consider a continuous spectra block of interest, *e.g.*, the U-NII bands $\mathcal{S} = [5.15, 5.925]$ GHz. Two spectrum monitors M_1 and M_2 watch over $\mathcal{S}$, which is divided into spectrum slices, numbered $\{s_L, s_L + 1, \cdots, s_H - 1, s_H\}$. Each spectrum slice of $\mathcal{S}$ can be in *idle* (in white) or *occupied* (colored) state, due to various spectrum activities.

An example. In Fig. 1a, $K = 3$ RAT's are allowed in this system: cellular (LTE/5G, RAT_1), IEEE 802.11 (WiFi, RAT_2), and IEEE 802.15 (Bluetooth, RAT_3). Among these, RAT_1 has the licensed 5G-NR FR1 bands [24] to itself, as indicated by LB_1, while its unlicensed U-NII bands UB_1 are shared with RAT_2, such that licensed-assisted LTE access co-exists with WiFi access [11], [23]. Meanwhile, the unlicensed ISM bands UB_3 are shared by RAT_2 and RAT_3. Then the spectra of interest $\mathcal{S} = \cup_{i=1}^{3} S_i$ is the union of these blocks.

Spectrum slice. Without loss of generality[3], we write $\mathcal{S}$ as interval $[s_L, s_H] \subset \mathbb{R}$, and further divide it into $\lceil \frac{s_H - s_L}{\Delta f} \rceil$ *spectrum slices* of width Δf, which is determined by:

(1) *Channel bandwidth* of $\{\text{RAT}_i\}_{i=1}^{K}$. There may not be a unified channel access scheme on $\mathcal{S}$ when $K > 1$. For instance, the U-NII bands can be accessed through LTE and WiFi. Under the former, the standard channel bandwidth are 1.4, 3, 5, 10, 15, and 20 MHz, while under the latter, the channel bandwidth ranges from 10 to 160 MHz. Further, an LTE channel is divided into resource blocks (180 KHz) that contain 12 sub-carriers, while each IEEE 802.11n channel (20 MHz) contains 52 sub-carriers that is of 312.5 KHz wide. Therefore, we choose the slice width Δf as a common divisor of all the channel bandwidths allowed by the K RAT's, such that a channel under each RAT_i contains k_i spectrum slices, where $k_i \in \mathbb{N}^+$ is a positive integer.

(2) *Resolution bandwidth* of monitoring devices. Due to the different sampling rates of commercial/prototype monitoring hardware, *e.g.*, 10 MS/s for USRP E310, and 2.4 MS/s for the low-cost SDR prototype designed in [14], which are constrained by their processing power (especially for FFT), and the stablizing time of the sweep-tune process, the resolution bandwidth of spectrum monitors are subject to various limits. Typically, it is set to be 1% to 3% of the channel bandwidth [10], [25] for observable results, but it is also required to be greater than 1 KHz to avoid overloading [25].

Based on these, a spectrum slice of width Δf will be used as the smallest[4] unit of spectrum trunk to be associated with an *access specification* and a observable *state*.

Access specifications. For each spectrum slice, administrator of the system specifies a legitimate way to access this slice, including allowed RAT, maximum transmitting power, maximum aggregated channel bandwidth, register/authentication procedure, and so on. For example, an 1 KHz slice in the 5G NR FR1 (LB_1 in Fig. 1a) can only be accessed through LTE/5G, with transmission settings specified in 3GPP technical specifications, *e.g.*, [24]. In this way, spectra block S is a database with $\lceil \frac{s_H - s_L}{\Delta f} \rceil$ items, against which monitors checks activities on each slice.

State. The state of a spectrum slice $i \in [1, \lceil \frac{s_H - s_L}{\Delta f} \rceil]$ (frequency range $[s_L + (i-1)\Delta f, s_L + i\Delta f)$) is the result of spectrum activities on this particular slice. Slice i is:

(1) *Idle*, when it is not occupied by any user, *e.g.*, the white slice $i = 4$ and slice $i = 5$ in time step $t = 1$ in Fig. 1b.

(2) *Rightfully occupied*, if it is accessed obeying the access specification. For instance, the blue slices $\{17, 18\}$ are rightfully occupied by a primary user (PU) at time $t = 2$; the green slice 8 and slices $\{12, 13\}$ (with ✔ marker) are accessed by authorized secondary users (SU) at time $t = 1$.

(3) *Illegitimately occupied*, if the occupant does not comply with the access specification of slice i. For instance, the purple slices $\{11, 14\}$ are beyond the designated spectrum slices of the aggressive PU at time $t = 3$; the red slices (with restriction sign) are used by an unauthorized SU during $t = 1$ to $t = 3$; the yellow slices $[1, 6]$ are jammed by an attacker emitting a high-power signal at $t = 2$. We refer to these illegitimate occupants as *spectrum culprits*, to be detected by monitors.

3. Spectra block $\mathcal{S}$ in an wireless overlay system may not form one single continuous interval, rather, it is the union of several non-overlapping continuous intervals, *i.e.*, $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots$. We focus on one of those intervals in this paper, for the simplicity of notation and understanding.

4. Note that Δf is not the frequency range that can be scanned by a monitor during a time step. For example, in Fig. 1b a monitor (red/blue box) can determine the states of 4 spectrum slices.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3032434, IEEE Transactions on Mobile Computing

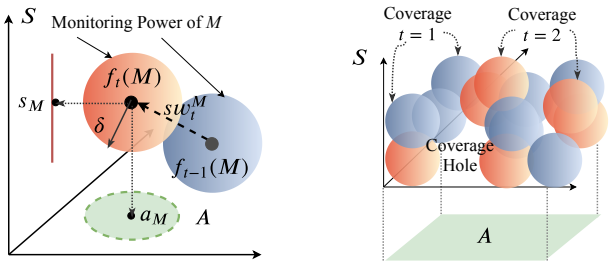


Fig. 2. $q(\delta)$-Monitoring power of a monitor is described as a δ-ball in space $X = \mathcal{S} \times \mathcal{A}$. Coverage of a strategy is the union of a collection of δ-balls centered at different assignment points, *e.g.*, the blue and orange shades on the right. The while the white space indicates the spectrum 'hole' that has not been monitored during the past two time steps.

2.1.2 The Spectra-location Space X

Consider the monitoring process of 1-D spectrum $\mathcal{S}$ over a closed 2-D geographical region $\mathcal{A} \subset \mathbb{R}^2$. Together, they compose a 3-D product space $\mathcal{S} \times \mathcal{A}$, referred to as the *spectra-location space* X. Then for any point $x \in X$, there exist projection maps $p_A : X \to \mathcal{A}$ and $p_S : X \to \mathcal{S}$ that identify the frequency and location of any point $x \in X$ respectively. In the product space X, the *spectra-location distance* d_{SA} between point x_i and point x_j is defined as the product metric induced by the Euclidean distance metrics d_S and d_A in both domains, that is,

$$d_{SA}(x_i, x_j) := ||d_S\left(p_S(x_i), p_S(x_j)\right), \frac{1}{\epsilon} d_A\left(p_A(x_i), p_A(x_j)\right)||_2, \quad (1)$$

where $||\cdot||_p$ denotes the p norm, and $\epsilon > 0$ is a scaling coefficient so that d_A and d_s are quantitatively comparable.

In this sense, the example in Fig. 1b is a special case, $\mathcal{A} = \{a\}$, in which only the spectrum domain $\mathcal{S}$ needs to be taken into consideration. But when the space domain $\mathcal{A}$ is sufficiently large, spectrum slices are annotated with locations, due to possible frequency re-use. In other words, spectrum activities take place in the product space X.

2.1.3 Surveillance Model

Denote $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$ as the set of m monitors in the system. During each time step, a monitor can only determine the states of adjacent slices [10], as illustrated by the boxes in Fig. 1b. Due to the attenuation of wireless signal over distance, such constraint also exists in the space domain $\mathcal{A}$ [10], hence the *monitoring power* definition.

Definition 1. ***($q(\delta)$-Monitoring Power)*** *For a monitor $M \in \mathcal{M}$ assigned at location $a_M^t \in \mathcal{A}$ and center frequency $s_M^t \in \mathcal{S}$ at time t, the monitoring power of monitor M is defined as a δ-ball centered at $s_M^t \times a_M^t \in X$, that is,*

$$Ball_\delta(s_M^t \times a_M^t) := \{x \in X | d_{SA}(s_M^t \times a_M^t, x) \leq \delta\}, \quad (2)$$

inside which spectrum activities (rightful/illegitimate occupancy) can be identified by monitor M with probability q.

Ball shape[5]. Parameter δ captures the *locality* of surveillance: the monitoring power of a single monitor is described as a *closed* δ-ball, illustrated in Fig. 2 (left), as a result of the trade-off between spectrum range (in spectrum distance d_S) and geographical range (in distance d_A), due to the limited sampling rate imposed by hard-ware constraints [14], [26]. To be more specific, the number of samples that a monitor can collect per unit time is *limited* [10], and these samples can be employed to cover either a larger bandwidth (large d_S) with a lower sensitivity, or a narrower bandwidth with a higher sensitivity. In other words, large d_S and large d_A can not be achieved simultaneously by a single monitor, hence the closed ball shape. For the most commonly-used energy detection method, *e.g.*, in [14], [15], lower sensitivity translates to a higher power threshold, resulting in a reduced detecting range in geographical domain, *i.e.*, small d_A.

Probabilistic outcome. Function $q : \mathbb{R} \to [0, 1]$ quantifies the reliability of results within the monitoring power, or equivalently the detection probability[6] of spectrum culprits. It has the following properties: i) q is a surjective; ii) q is non-decreasing in $\mathbb{R}$; iii) we can define its inversion $q^{-1} : [0, 1] \to \mathbb{R}$ as $q^{-1}(y) := \sup_{x>0}\{q(x) = y\}$, so for any required reliability $y \in [0, 1]$, there exists a critical radius $\delta^* = q^{-1}(y)$, above which the monitoring results are not acceptable. Consequently, if a point $x \in X$ is covered by the $q(\delta)$-monitoring power of k monitors, illegitimate occupancy at this point x can be detected with a higher probability, that is, $1 - [1 - q(\delta)]^k$.

Parameters. By fine-tuning function $q(\cdot)$, radius δ, and parameter ϵ in Eq. (1), a variety of monitoring techniques can be depicted by this δ-ball model. These parameters are determined by hardware performances, including sensitivity, noise floor, input range, and the detection algorithm. Radius δ refers to $\delta_* = q^{-1}(1)$ that can guarantee a fully reliable detection result, if not explicitly specified hereafter.

2.1.4 Exploit Model

Recall that a spectrum culprit at time t is defined as the occupant of a spectrum slice that does not comply with the access specifications, as exemplified in Fig. 1b. The gist of spectrum exploit is that, a spectrum culprit $R \in \mathcal{R}$ located at $a_R \in \mathcal{A}$, illegitimately occupies one or multiple spectrum slices, denote as $S_R \subset \mathcal{S}$, at time t. As a result, R leaves a 'mark' $R_t = S_R \times a_R \subset X$. The wider S_R is, the larger the 'mark', and the more detectable R becomes. We make following assumptions about the culprit:

(1) *Narrow* S_R: We consider spectrum culprits that are most difficult to detect as the worst-case scenario. In this case, S_R shrinks to a point $\{s_R\} \in \mathcal{S}$, such that $R_t \in X$.

(2) *Constant Presence*: We assume culprits stays in the

5. Technically, some spectrum slices will be divided by the sphere, excluding which the monitoring power is not a smooth ball. We still refer to the monitoring power as δ-ball for the ease of comprehension. The divided slices is not a problem, because: i) the slice width Δ_f is oftentimes much thinner than the width $s_H - s_L$ of the spectra block $\mathcal{S}$ (*e.g.*, the 775 MHz U-NII frequency bands); ii) the δ-balls are forced to overlap in the space-tessellation step (Sec. 3. A) such that there will not be coverage gaps.

6. For any radius δ, it is more probable to determine whether a spectrum slice at a location is occupied or not ($q_c(\delta)$), than determining whether the occupancy is legit ($q_c(\delta)$). Consequently, $q_c(\delta)$ for occupancy measurement is greater than or equal to that for culprit detection $q_d(\delta)$. We set $\delta = \max\{\delta > 0 \mid q_c(\delta) = 1\}$, and $q = q_d(\delta)$, such that occupancy measurement is accurate, while culprit detection is not.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3032434, IEEE Transactions on Mobile Computing

system, and continues its misbehavior[7] throughout time $\mathcal{T}$.

Detect a culprit at time t. When the exploit mark R_t overlaps with the monitoring power of some monitors, *i.e.*, $\exists M_i \in \mathcal{M}$ such that $R_t \cap Ball_\delta(f_t(M_i)) \neq \phi$, culprit R is *detectable* at time t with probability at least $q(\delta)$.

Exploit sequence. Over a period of time, the exploit marks constitute an *exploit sequence* $\{R_t\}_{t\in\mathcal{T}}$ of culprit R. The detailed exploit *pattern* of spectrum culprits, *i.e.*, how a spectrum culprit $R \in \mathcal{R}$ assigns its exploit sequence, can be either oblivious or adversarial, depending on its learning capability, and will be discussed in Sec. 3.B.

2.1.5 Switching Model

Despite their different roles, all the entities, *i.e.*, normal users (denoted as $\mathcal{U}$), spectrum culprits ($\mathcal{R}$), and monitors ($\mathcal{M}$), are all wireless devices that are capable of moving and tuning. Both actions will result in a relocation of devices in the spectra-location space X between time steps, which we refer to as a *switching* sw_t^Y of device $Y \in \mathcal{U}\cup\mathcal{M}\cup\mathcal{R}$, that is, a move of Y from point $Y_{t-1} \in X$ to point $Y_t \in X$ at the end of time step $t-1$. For example, in Fig. 2 left, the switching sw_2^M corresponds to the relocation and tuning of monitor M between time step $t=1$ and $t=2$, resulting in a change of coverage, as shown in Fig. 2 right. This common action is also constrained by time, energy or other kind of cost, as opposed to the assumptions in [14], [19]. This constraint is indeed a design concern in SAS processes, to capture which we define the *switching capacity*.

Definition 2. *(**Switching Capacity**) Let $Y_t \in X$ denote the location (in space X) of device Y at time t, the switching capacity α_Y of Y is defined as the maximum distance in X, that device Y can switch over by one action in a time step, that is,*

$$\alpha_Y := \sup_{t\in\mathcal{T}} \{d_{SA}(Y_t, Y_{t+1})\}. \tag{3}$$

Device Y is referred to as an α_Y-monitor or α_Y-culprit.

Dedicated SAS monitors. For a dedicated monitor, a switching is composed of physical movement and/or tuning. Consequently strategy design is restricted by a quantitative switching cost, including time, energy, budget *etc*, that is a function of the switching distances $d_{SA}(\cdot,\cdot)$.

Crowd-source SAS monitors. Switching actions in a crowd-source scenario are merely changes of surrogate monitors. Therefore, if immediate communication among participants is guaranteed, or there exists a central controller capable of timely coordination, switching will not be constrained, *i.e.*, $\alpha_M = \infty$; otherwise for the case of distributed control, which relies on local wireless communication, switching is not possible beyond the communication range of monitors.

7. Culprits such as deliberate jammers or unregistered users, are likely to misbehave across time, while it is also possible that a culprit misbehaves occasionally, *e.g.*, hogging spectrum trunks without proper authorization at one time step t, and then behaves nicely during the next few steps, or even leave the system and never to find again. For the latter case, it is difficult and not meaningful to study performance of SAS strategies when culprits come-and-go spontaneously. But it is also possible that once the culprit convicted its first crime, some attributes of the culprit, *e.g.*, spectrum fingerprint [27], are known to the SAS monitors, *e.g.*, through reports of others, such that a match can be found even if the culprit has stopped its misbehavior by now.

2.2 SAS Strategy and Metrics

At the beginning of time step t, each monitor $M_i \in \mathcal{M}$ is assigned to a spectra-location point $f_t^m(M_i) \in X$, through an *assignment* map $f_t^m : \mathcal{M} \to X^m$, *e.g.*, f_1 and f_2 in Fig. 2. Allowing time t to proceed in $\mathcal{T}$, assignment points of the m monitors constitute a *strategy*.

Definition 3. *(**Strategy**[8] f_t^m) A strategy $\{f_t^m\}_{t\leq T}$ is a sequence of assignments during time interval $[1,T] \subset \mathcal{T}$, which are carried out by the m monitors in set $\mathcal{M}$, under their (switching) capacity constraints $\{\alpha_{M_i}\}_{M_i\in\mathcal{M}}$. During time step t, monitors in set $\mathcal{M}$ can scan $C(f_t^m) = \bigcup_{M_i\in\mathcal{M}} Ball_\delta(f_t^m(M_i))$ en masse, which is referred to as the (surveillance) coverage of assignment f_t^m. Sweep-coverage of a strategy $\mathcal{C}(f)$ is then the union of sequence $\{C(f_t^m)\}_{t\in[1,T]}$ across time interval $[1,T]$.*

Performance metrics. Recall the two objectives of the SAS function, that is, spectrum occupancy measurement and spectrum culprit detection. The former urges for a quick sweep-scan of the entire spectra-location space X, *i.e.*, minimizing the time needed to satisfy the coverage goal $X \subset \mathcal{C}(f)$, such that spectrum (occupancy) status can be timely recorded and updated to users. The latter requires effective detection of spectrum culprits, such that the time that an undetected culprit illegitimately occupies spectrum slices can be reduced. For instance, in the special case scenario ($X = \mathcal{S}\times\{a\}$) illustrated in Fig. 1b, the entire X is sweep-covered at $t=3$, and the unauthorized SU (culprit) exploited the system for two time steps before its detection at $t=3$. In other words, the efficacy of a SAS strategy can be quantitatively evaluated and fairly compared through the following two metrics in the time domain, with respective to the coverage and detection goals.

Definition 4. *(**Coverage Time** T_f^m**, Detection Time** $\tau_R(f^m)$**)** Under strategy $\{f_t^m\}_{t\in T}$, the coverage time is defined as the first time that the sweep-coverage $\mathcal{C}_T(f^m)$ contains every point in space $X = \mathcal{S}\times\mathcal{A}$, that is,*

$$T_f^m := \min\{T \in \mathcal{T} \mid x \in \mathcal{C}_T(f^m),\ \forall x \in X\}. \tag{4}$$

The detection time of a culprit R with exploit sequence $\{R(t)\}_{t\in T}$, is defined as the first time that culprit R can be identified by any of the m monitors, that is,

$$\tau_R(f^m) := \min\{t \in \mathcal{T} \mid \sum_{i=1}^{m} \mathbb{1}_{R_t\in Ball_\delta(f_t^m(M_i))} D_i \geq 1\}, \tag{5}$$

where detection outcome D_i is a Bernoulli r.v. with mean q.

The detection time in Eq (5) can be further simplified to

$$\tau_R(f^m) := \min\{t \geq 1 \mid R(t) \in C(f_t^m)\}, \tag{6}$$

if surveillance (detection) result is fully reliable, *i.e.*, radius of the monitoring power $\delta \in \{\delta^* > 0 \mid q(\delta^*) = 1\}$.

In this paper, we focus on these two metrics from the time aspect, for delay-sensitive applications such as DSA-related spectrum database construction. We note that there are more aspects to consider in SAS, including energy consumption, expenses of mounting/recruiting monitors,

8. Superscript m in f_t^m denotes the number of monitors, while subscript t denotes time. A second subscript may be added to differentiate strategy types, *e.g.*, $f_{S,t}^m$ for deterministic strategy. Any of the three (m, time, and type) may be omitted, when no confusion is raised.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3032434, IEEE Transactions on Mobile Computing

predictability to 'smart' culprits, and so on. In addition to these macroscopic measures, microscopic performance metrics, such as the turn-around time between scans over specific frequency ranges or geographical regions, can also be evaluated for refined strategy studies.

2.3 The SAS Strategy Design Problem

With the system model described in this section, this paper studies the SAS strategies for a set of m α_M-monitors, to achieve the sweep-coverage and culprit detection goals in the spectra-location space X. Specifically, we aim to design strategies $\{f_t^m\}_{t\in\mathcal{T}} \in \{X^m\}^{\mathcal{T}}$ for dedicated and crowd-source surveillance scenarios, and examine their *efficacy* by answering the following questions:

1) What is the the coverage time T_f of the designed strategy f^m, by which time spectra-location space X is sweep-covered, *i.e.*, $X \subset \mathcal{C}(f^m)$?
2) Under the strategy f, what is the detection time $\tau_R(f^m)$ of an α_R-spectrum culprit $R \in \mathcal{R}$ with the exploit sequence $\{R_t\}_{t\in[1,\mathcal{T}]}$?

3 A Two-step Solution

In essence, designing a SAS strategy $\{f_t^m\}_{t\in\mathcal{T}} \in \{X^m\}^{\mathcal{T}}$ is equivalent to finding a sequence of assignment points in the spectra-location space X, for every monitor $M \in \mathcal{M}$ at every time instant t, subjecting to the switching capacity constraint α_M. Two major challenges arise in this process: First, for every time t, the solution space X^m is of infinite size, which hinders both the analysis approach and the search-based experiment approach. Second, switching actions of monitors, *i.e.* the tuning (a move in the spectrum domain $\mathcal{S}$) and/or relocation (a move in the geographical space domain $\mathcal{A}$), are constrained in range, due to the switching cost they incur, which further complicates the problem. To overcome these challenges, we propose a two-step solution: first, the continuous strategy space $\{X^m\}^{\mathcal{T}}$ is reduced to a discrete and *finite* space $\{V^m\}^{\mathcal{T}}$ through space-tessellation; then any surveillance strategy is formulated as a *walk* on the graph, whose edges illustrate possible switching path of monitors. In this way, SAS as a global activity is transformed into a chain of individual actions, *i.e.*, switching (walking) of monitors and culprits, such that design of SAS strategies becomes tractable.

3.1 Space Tessellation: Reducing the Strategy Space

Driven by the sweep-coverage and culprit detection objectives, the assignment points $\{f_t(\mathcal{M})\}_{t\in[1,T_f]}$ of a good SAS strategy should have the following properties:

(1) *Least points*. To timely update the spectrum occupancy data, monitors are expected to sweep-scan the entire space X as quickly as possible, that is, for a strategy $\{f_t\}_{t\in[1,T_f]}$ to achieve the coverage goal $X \subset \cup_{t=1}^{T_f} C(f_t)$ with as few $(m * T_f)$ assignment points as possible.

(2) *Minimal overlapping*. To quickly detect culprits, every assignment f_t is expected to cover as much space (large $C(f_t)$) as possible, which translates to a minimal overlapping of monitoring powers during every assignment[9].

9. Though overlapping monitoring power permits a higher detection probability inside the δ-ball than that of a single coverage, it is not necessary when the detection probability q is sufficiently high.

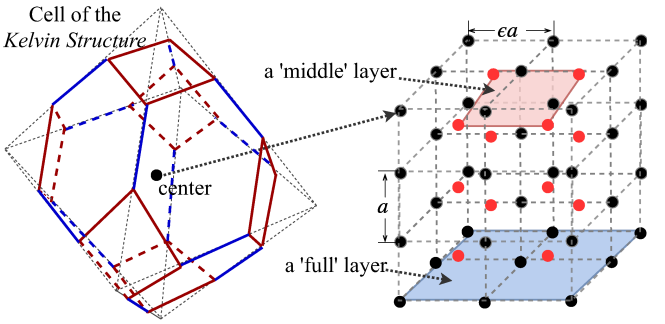


Fig. 3. Each cell in the Kelvin structure is a truncated octahedron (left) with eight hexgonal faces and six square faces. The discrete assignment space V of a rectangular region $\mathcal{A}$ is composed of cell centers (right), arranged in inter-leaving 'full' (black) and 'middle' (red) layers.

These requirements can be jointly satisfied if the continuous space X is divided into a minimum number of non-overlapping *cells*, each covered/contained by a δ-ball, *i.e.* the monitoring power of a monitor, and every assignment map f_t takes value from the set V of cell centers, instead of the entire X. In this way, the first step becomes a *tessellation* problem of space X.

3.1.1 Solution to the Space-Tessellation Problem

Space tessellation, or honeycomb, in the 3-D space X, refers to the close packing of 3-D cells without overlaps or gaps.

Choices of cells. A cell in the tessellation can be regarded as the non-overlapping part of a δ-ball (monitoring power), and comes in various forms in the solution to this classic problem, *e.g.*, cube, hexagon-prism, tetrahedra, etc. The higher volume-efficiency ρ of a cell, that is, the volume ratio of the cell over its insubscribed δ-ball, the more efficient the form of the cell, due to the less overlap between adjacent monitoring power (δ-balls). It would be ideal to fully utilize δ-balls to fill the space. However, direct packing of solid balls always leaves gaps, *i.e.*, spectrum holes in the sweep-coverage, which can be eliminated by pushing the 'elastic' balls into each other. Meanwhile each 'squeezed' balls (the resulting cells) should be inscribed to a δ-ball, *i.e.* the maximum distance of any two points on the cell surface is required to be smaller than 2δ.

Kelvin structure. The space tessellation problem is closely related to the *Kelvin problem* [28], which aims to find the most efficient bubble wrap form to fill a space with the least surface area. In the Kelvin problem, efficiency is quantified by the *isoperimetric quotient* [29], and the ball shape (as the monitor power in our model) has the highest isoperimetric quotient value of 1. Therefore, the most efficient cell form in the tessellation problem is the one that has the highest isoperimetric quotient. In this sense, the best solution for wide (large $|\mathcal{S}|$) spectrum is the *Kelvin structure* (with isoperimetic quotient value 0.757), whose cells are truncated octahedrons (Fig. 3 left), arranged in a layered manner (Fig. 3 right). The truncated octahedron cell has the highest volume-efficiency ρ compared to other forms, as shown in Table 1. Centers of these cells correspond to assignment points composing the discrete and *finite assignment space* V, whose size $n = |V|$ can be obtained from the following proposition.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3032434, IEEE Transactions on Mobile Computing

TABLE 1
Volume Efficiency of Different Cell Forms

Cell Form	Volume Efficiency ρ	Isoperimetric Quotient	Size of Assignment Space $\lvert V\rvert$
Cube	$\geq \frac{2}{\sqrt{3}\pi} \simeq 0.368$	$\frac{1}{216}$	$\lceil \frac{\sqrt{3}\lvert\mathcal{S}\rvert}{2\delta} \rceil \cdot \lceil \frac{3\lvert\mathcal{A}\rvert}{8(\epsilon\delta)^2} \rceil$
Hexagon-prism	$\geq \frac{3\sqrt{3}}{2}\alpha\sqrt{(1-\alpha^2)} \leq 0.585$	$\frac{3\sqrt{3}b}{2(3\sqrt{3}+12b)^3}$, where $b = \sqrt{\frac{\lvert\mathcal{S}\rvert^2}{4\delta^2-\lvert\mathcal{S}\rvert^2}}$	$\frac{2\lvert\mathcal{A}\rvert}{3\sqrt{3}(1-\alpha^2)(\epsilon\delta)^2} \rceil$
Truncated octahedron	$\geq \frac{24}{5\sqrt{5}\pi} \simeq 0.683$	0.757	$\lceil \frac{5\sqrt{5}\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{16\epsilon^2\delta^3} \rceil$

Proposition 1. ***(Size of the Assignment Space $\lvert V\rvert$)*** *When the spectrum block $\mathcal{S}$ is narrow ($\lvert\mathcal{S}\rvert = 2\alpha\delta$, $0 < \alpha << 1$), the size of the assignment space $n = \lvert V\rvert$ can be determined by tessellation with hexagon-prism cells, i.e.,*

$$n_{hex} = \lceil \frac{2A}{3\sqrt{3}(1-\alpha^2)(\epsilon\delta)^2} \rceil, \tag{7}$$

where A denotes the area of region $\mathcal{A}$, δ corresponds to the monitoring power and ϵ is the scaling coefficient in Eq. (1)*. Otherwise, size n is achieved by tessellation with truncated octahedron (Kelvin structure) cells, and*

$$n_o \geq \lceil \frac{5\sqrt{5}\lvert\mathcal{S}\rvert A}{16\epsilon^2\delta^3} \rceil. \tag{8}$$

Further, if the geographical region $\mathcal{A}$ is rectangular,

$$n_o \geq \lceil \frac{\sqrt{5}\lvert\mathcal{S}\rvert}{4\delta} \rceil \cdot \lceil \frac{5A + 2\sqrt{5A}\delta(3-2\epsilon) + 4\delta^2(1-\epsilon)^2}{8(\epsilon\delta)^2} \rceil, \tag{9}$$

where $e = \frac{2}{\sqrt{10}}\delta$ is the edge length of cells.

Proof of Proposition 1 can be found in Appendix A.

3.1.2 Exploit Patterns of Culprits in Assignment Space V

By space-tessellation, sweep-coverage of X is guaranteed as long as the assignment maps $\{f_t\}_t$ are jointly surjective on the *finite* assignment space V of cell centers. In other words, it is enough for monitors ($\mathcal{M}$) to switch among assignment points in V, which greatly reduces the size of the solution space (from ∞ to $\lvert V\rvert^m$) for every time step. For the ease of notation and discussion, we can also restrict the range of exploit points to V, because any point x in X can be uniquely mapped to a cell with its center v_x in V, and the probability of a spectrum culprit $R \in \mathcal{R}$ being detected at $R_t \in X$ is the same as that at $R_t = v_x$. In this context, we categorize the exploit patterns of spectrum culprits, *i.e.*, how a spectrum culprit $R \in \mathcal{R}$ determines its exploit point R_t for the next time step, with respect to its learning capabilities.

Definition 5. ***(Persistent Culprit R_p)*** *A persistent culprit $R_p \in \mathcal{R}$ refers to a culprit, whose exploit pattern does not change over time, that is, $\{R_t\}_t$ is composed of i.i.d. r.v.'s R_t^p, all distributed with PMF $g_{R_p}(v)$, where $v \in V$.*

Persistent culprit R_p can describe a variety of exploiting strategies with different PMF $g_{R_p}(v)$. For instance, as shown in Table 2, R_s is a stationary culprit that can only access a selective range of frequencies; R_{sd} corresponds to a DSA-enabled stationary culprit; and R_{md} is a mobile DSA-enabled culprit that can move in region $\mathcal{A}$.

As opposed to oblivious persistent culprits, machine learning-assisted radio access technology [12], [13] allow sophisticated culprits to steer the game toward their benefit, by actively dodging monitors [19]. The intuition behind adversarial culprits is that, once a SAS strategy is known[10] by a culprit R_a, R_a can then switch to points that are less probable to be monitored in the next time step.

Definition 6. ***(Adversarial Culprit R_a)*** *An adversarial culprit $R_a \in \mathcal{R}$ is a culprit with prior knowledge of the current strategy $\{f_t^m\}_{t\in\mathcal{T}}$, that is, R_a knows the set of probabilities $\{v \in \bigcup_{M_i\in\mathcal{M}} f_t^m(M_i)\}_{v\in V}$ ahead, and determines its current exploit point R_t^a with PMF $g_{R_a}^t(v) = \frac{1}{\lvert\text{Void}(t)\rvert}$, for point $v \in \text{Void}(t)$, where $\text{Void}(t) = \arg\min_{v\in V} \mathbb{P}\Big(v \in \bigcup_{M_i\in\mathcal{M}} f_t(M_i)\Big)$.*

Note there is no switching constraint in Def. 5 and Def. 6, which describes the most powerful culprits in terms of switching, *i.e.*, $\alpha_R = \infty$. For culprits with switching capacity $\alpha_R < \infty$, the range of R_{t+1} will be restricted to a smaller subset $N(R_t) = \{v \in V \mid d_{SA}(v, R_t) \leq \alpha_R\}$ of V, with the selecting probability of a point $v \in N(R_t)$ recalculated as $\mathbb{P}(N_{t+1} = v) = \frac{g_R(v)}{\sum_{u\in N(R_t)} g_R(u)}$.

Through space-tessellation, the continuous (and hence infinite) strategy space $\{X^m\}^{\mathcal{T}}$ is reduced to discrete and finite $\{V^m\}^{\mathcal{T}}$ for both monitors and culprits. Next, we discuss how to incorporate switching capacity constraints (of monitors and culprits) in the design of SAS strategies.

3.2 Graph Walk: A Chain of Switching Actions

Over the discrete time span $\mathcal{T}$, any SAS process is now a chain of switching actions in the assignment space V. Recall that a switching sw_t^Y is a relocation (tuning in $\mathcal{S}$ and/or movement in $\mathcal{A}$) of a device $Y \in \mathcal{M} \cup \mathcal{R}$ (monitor or culprit[11]) from time $t-1$ to t, whose range is upper-bounded by the switching capacity α_Y. Next, we discuss switching actions from range (how far) and time (how quick) aspects, to formulate the SAS process into a graph walk problem.

3.2.1 Range Aspect (Switching Capacity)

The switching range refers to the maximum distance d_{SA} over which one switching action is possible. A switching action induces cost, in the form of time, energy, budget, and so on, and may hence be constrained. For instance, to dedicated monitors fully controlled by the SAS system

10. We consider the most powerful culprit (with full knowledge of a strategy) as an extreme case to examine the performance of an SAS system against compromised strategies. A weaker culprit can at least observe the long-term visiting probability of any point $v \in V$ as knowledge. In other cases, SAS strategies are required to be disclosed. For instance, in a crowd-source scenario, any participants will need to acquire information of the strategy, which increases the risk of a strategy being leaked to culprits.

11. Switching actions of normal users in $\mathcal{U}$ do not have any impact on the performance of SAS strategies, and are hence not considered here.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3032434, IEEE Transactions on Mobile Computing

TABLE 2
Persistent Spectrum Culprits and Their Detection under Deterministic Strategy f_S^m

Type	Description	PMF $g_{R_*}(\cdot)$	Expected Detection Time $\mathbb{E}(\tau_*(f_S))$
R_s	Stationary fix-frequency culprit	$g_{R_s}(v) = \mathbf{1}_{v_s}(v),\ \forall v \in V$	$\frac{T_s}{2}$, where T_s is discussed in Sec. 4.2.
R_{sd}	Stationary DSA-enabled culprit	$g_{R_s}(v) = \begin{cases} \frac{1}{\lvert V_{sd}\rvert}, & \text{if } v \in V_{sd}, \\ 0, & \text{otherwise.} \end{cases}$	$\leq \frac{n}{m} \simeq T_s$
R_{md}	Mobile DSA-enabled culprit	$g_{R_{md}}(v) = \frac{1}{n},\ \forall v \in V$	$\frac{n}{m} \simeq T_s$

operations, switching is completed via physical movement and/or tuning of individual monitors, so switching cost scales with spectra-location distance d_{SA}. In this case, we need a quantitative metric to accurately measure the switching cost for strategy design. This SAS scenario is discussed in Sec. 4, for which a low switching-cost strategy is proposed for dedicated monitors.

Switching in a crowd-source scenario is a change of surrogate monitors, that is, wireless devices (spontaneously) participating in the SAS process. In this case, the switching cost may not scale with the spectra-location distance. Instead, a switching between any two assignment points v_x and $v_y \in V$, can either be possible with a fixed amount of cost (*e.g.*, time and coordination budget), or impossible during one time step. To be more specific, when immediate communication is guaranteed among all participants, a 'handover' between any two monitoring surrogates (switching) will be possible in one time step, *i.e.*, $\alpha_M = \infty$; otherwise for distributed crowd-source monitoring that relies on short-range wireless communication to coordinate, any switching action is only possible between two monitors within their communication ranges, *i.e.*, $\alpha_M < \infty$. For crowd-source monitoring scenarios under unlimited ($\alpha_M = \infty$) and limited ($\alpha_M < \infty$), strategy design is discussed in Sec. 5 and Sec. 6, respectively.

3.2.2 Time Aspect (Switching Rates)

In addition to range, the rate of switching, that is, how many switching actions can be done in one time step, is also constrained by the hardware. Intuitively, a faster-switching culprit will be more difficult to catch, due to the shorter time that the culprit remains in the monitoring power of any monitors. It is not an issue in coverage time, which is irrelevant to switching actions of culprits. So, we examine its impact on the detection time.

Consider spectrum monitors with $q(\delta)$-monitoring power, that is, when a culprit shows up in the cell assigned to a monitor (referred to as *co-location*), the probability that it is detected by that monitor during one time step is q. Let $q \cdot p(s)$ ($s \in [0,1]$) denote the detecting probability when the co-location time s is less than one full time step, where the non-decreasing function $0 \leq p(s) \leq 1$ captures the attenuated detection probability due to the decreased co-location time, and has the property of $p(0) = 0$, $p(1) = q$.

Lemma 1. ***(Detection of a Faster Culprit)*** *Suppose culprit R_1 differs from R_2 only in switching rates: R_1 can switch $k \in \mathbb{N}^+$ times during one time step, while R_2 and monitors can switch only once. The detection time of R_1 is stochastically dominated by that of R_2, that is, for any strategy f,*

$$\tau_{R_1}(f) \overset{d}{\leq} \tau_{R_2}(f), \tag{10}$$

when the following condition holds:

$$p(\frac{1}{k}) \geq \frac{1 - [1 - q \cdot \mathbb{P}\big(R^2(t) \in C(f_t^m)\big)]^{\frac{1}{k}}}{q \cdot \mathbb{P}\big(R^2(t) \in C(f_t^m)\big)}. \tag{11}$$

Proof of Lemma 1 can be found in Appendix B.

The condition in Eq. (11) easily holds when probability $\mathbb{P}\big(R^2(t) \in C(f_t^m)\big)$ is small, *i.e.*, when the size of the assignment space $|V|$ is sufficiently large, such that it is difficult for a culprit to co-locate with any monitor. Otherwise, when the assignment space V contains few assignment points, though catching the faster culprit R_1 takes more time than the slower R_2, the expected detection time can be derived as $\mathbb{E}(\tau_{R_1}) = \frac{1}{kq \cdot p(\frac{1}{k}) \mathbb{P}(R_2 \in C(f_t^m))}$, which will not be large, if function value $p(\frac{1}{k}) \geq \frac{1}{k}$. Consequently, the difference between the detection time of R_1 and R_2 will be small. Based on this observation, it is reasonable to assume both culprits and monitors switch once in every time step.

3.2.3 Graph Walk on (G_M, G_R)

Accounting switching capacity, the assignment space V is more than a set of points, rather, a subspace that inherits the d_{SA} metric from space X. Then, this subspace gives rise to a structure that incorporates the possibility of switching actions, *i.e.*, a composite graph of:

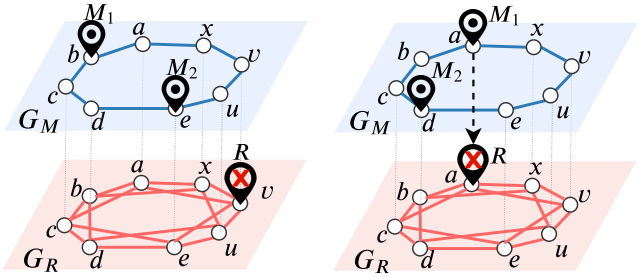
(1) **Monitoring subgraph** $G_M = (V, E_M)$, in which an edge $(u, v) \in E_M$ exists, if and only if $d_{SA}(u, v) \leq \alpha_M$. An arbitrary strategy $\{f_t^m\}_{t \in \mathcal{T}}$ can be seen as a joint *walk* by $m = |\mathcal{M}|$ monitors on the monitoring subgraph G_M.

(2) **Exploiting subgraph** $G_R = (V, E_R)$, in which the edges are constructed the same way under the switching capacity α_R of the culprit. Then, the exploiting sequence $\{R_t\}_t$ of R, which contains the assignment points exploited by culprit R, also corresponds to visited vertices of a walk on the exploiting subgraph $G_R = (V, E_R)$.

Graph G_R and G_M have the same vertex set V, and are both sub-graphs of K_n, *i.e.*, the complete graph with $n = |V|$ vertices, which corresponds to SAS scenario of unlimited switching capacities ($\alpha_M = \alpha_R = \infty$). The SAS process, particularly culprit detection, then becomes a graph walk on the composite graph $G = (G_M, G_R)$, in which culprit R is first detected when R and any of the monitors in $\mathcal{M}$, co-locate at an assignment point, *i.e.*, meet on a vertex in G.

An example. Fig. 4 illustrates a graph walk on the assignment space $V = \{a, b, c, d, e, u, v, x\}$ for two time steps. Two monitors $\mathcal{M} = \{M_1, M_2\}$ (marked with black dot in the center) walk on the monitoring subgraph G_M, whose edges E_M are shown in blue lines, while a culprit R (marked with red '$\times$' in the center) walks on the exploiting subgraph G_R, whose edges E_R are shown in red lines. By time $t = 2$, the sweep-coverage $\mathcal{C}_t = \{a, b, c, e\}$ has not

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3032434, IEEE Transactions on Mobile Computing



(a) $t = 1$ (b) $t = 2$

Fig. 4. SAS process as a graph walk: Two monitors (M_1 and M_2, blue dots) and one culprit (R, red dot) move on composite graph (G_M, G_R), in which the monitoring subgraph $G_M = (V, E_M)$ (blue edges) is sparser than the exploiting subgraph $G_R = (V, E_M)$ (red edges), due to the weaker switching capacity $\alpha_M < \alpha_R$. The culprit R is detected at time $t = 2$ when it co-locates with monitor M_1 at assignment point a.

covered the assignment space V yet, indicating the coverage time $T_f > 2$. The detection time $\tau_R(f) = 1$, because monitor M_1 meets with the culprit R_t at assignment point $a \in V$, during time step $t = 2$.

Formulating the SAS process into a graph walk makes strategy design more tractable, in the sense that the strategy space is now discrete and finite, such that both the theoretic and simulation approaches are viable. Under this formulation, we discuss strategy design and performance evaluation for the dedicated and crowd-source monitoring scenarios, in the following Sec. 4 and Sec. 5, respectively.

4 Deterministic Strategies for Dedicated Monitors

Dedicated monitors refer to the specialized monitoring equipment mounted on towers, drones, vans, etc. which are widely used by governmental and commercial agents, *e.g.*, FCC, NTIA, and AT&T, to collect spectrum measurement data [6]. For DSA-enabled systems relying on dedicated monitors, *deterministic* monitoring strategies, in which monitors traverse a predetermined route to sweep-scan the spectra-location space, are the sensible choice, due to its simpleness, *e.g.*, measurement taken in [10].

Design concern. For such strategies, the *switching cost*, in the form of time, energy or budget, is the key concern in deploying dedicated monitors. The reason behind this is that, switching cost, induced by tuning (in spectrum domain $\mathcal{S}$) and movement (in space domain $\mathcal{A}$), scales with distances in both domains, so it is essential to optimize the strategy for a reduced cost, given that all the monitors are under the control of the SAS function. Therefore, we first define a comprehensive switching cost metric, based on the optimization of which, we propose the low-cost deterministic SAS strategy f_S.

Definition 7. ***(Switching Cost** Γ**)*** *The switching cost from point x_i to $x_j \in X$, is defined as the sum of tuning cost and relocation cost, that is,*

$$\gamma(x_i, x_j) := \beta_S d_S(p_S(x_i), p_S(x_j)) + \beta_A d_A(p_A(x_i), p_A(x_j)), \tag{12}$$

where β_S and β_A are cost coefficients for tuning and relocation respectively. The cost of strategy $\{f_t^m\}_{t=1}^{T_f}$ is then

$$\Gamma_f^m := \sum_{t=2}^{T_f^m} \sum_{i=1}^{m} \gamma(f_t^m(M_i), f_{t-1}^m(M_i)), \tag{13}$$

where T_f^m is the coverage time defined in Eq. (4).

Settings of β_S and β_A. Switching cost Γ_f can be applied to describe time, energy, and budget. For example, the cost coefficients can be set to $\epsilon\beta_A >> \beta_S$ to describe the switching time, since the time it takes a radio head to tune to a different center frequency is approximately 1 ms [30], during which the physical movement of any mobile device is negligible. It could also be the case that re-configuring the center frequency of a radio head is more expensive (in terms of budget) than physical movements, *e.g.*, specialized devices with narrow frequency ranges, such that $\epsilon\beta_A < \beta_S$. Accounting for switching cost between assignment points in space V, the resulting monitoring sub-graph G_M becomes a weighted complete graph K_n, in which the weight of each edge reflects the switching cost along that edge.

4.1 Low Cost Deterministic Strategies f_S

A good deterministic strategy for dedicated monitors is a strategy $\{f_{S,t}^m\}_{t\in\mathcal{T}}$, whose total switching cost Γ_f^m is minimized in the strategy space $\{V^m\}^{\mathcal{T}}$. In this sense, finding an optimal strategy $\{f_{S,t}^m\}_{t=1}^{T_s}$ is equivalent to finding m vertex-disjoint 'shortest' paths (in terms of switching cost) that jointly cover the assignment space V by time $T_S^m = \lceil \frac{n}{m} \rceil$. This problem is actually an open-path multi-depot multi-travailing salesmen problem (MD-MTSP), which is known to be NP-hard [31].

Solution to this NP-hard problem. Observing the structure of assignment space V (as shown in Fig. 3 right) over a rectangular region $\mathcal{A}$, we present upper bounds on the minimum switching cost Γ_{min}^m for small m, as an extension of our prior work [21], which accounts for all cost coefficient settings. Let L and D denote the number of assignment points along the length and width of $\mathcal{A}$, and H denote the number of assignment points along the spectra axis $\mathcal{S}$. For instance, the assignment space V in Fig. 3 has $L = D = 3$ and $H = 4$. Then total number of points $n = |V| = LDH + (L-1)(D-1)(H-1)$. Let $\gamma_1 = \beta_S a$, $\gamma_2 = \beta_A \epsilon a$ and $\gamma_3 = \frac{a}{2}(\sqrt{2}\beta_A\epsilon + \beta_S)$ denote the switching cost of type-1,2, and 3 edges, which are the single-hop paths to the nearest point if only one switching action is allowed in spectrum (alone), in space (alone), and in both domains, respectively.

Theorem 1. ***(Min. Switching Cost)*** *For a set of $m \geq 3$ monitors on the assignment space V with parameters L, D and H, the switching cost Γ_{min}^m can be upper bounded by*

$$\Gamma_{min}^m \leq \begin{cases} \Gamma_*^1 - A_*(m-1), & \text{if } m = 2k+1, \\ \Gamma_*^2 - A_*(m-2), & \text{if } m = 2k+2, \end{cases} \tag{14}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3032434, IEEE Transactions on Mobile Computing

where $k \in \mathbb{N}^+$, $* = f$ *if* $\beta_S \geq \frac{2-\sqrt{2}}{2}\beta_A\epsilon$, *and* $* = g$ *otherwise. Quantities* Γ^1_*, Γ^2_* *and* $A_*(\cdot)$ *are determined by*

$$\begin{aligned}
\Gamma^1_* &= K_1\gamma_1 + K_2^*\gamma_2 + K_3^*\gamma_3,\\
\Gamma^2_f &= \Gamma^1_f - (2D-1)\gamma_2 + \gamma_3 + (\frac{H}{2}\gamma_1 + \gamma_2)e(L),\\
\Gamma^2_g &= \Gamma^1_g - 2\gamma_2 - \gamma_1 + 2\gamma_3,\\
A_*(x) &= \min\{K_1, x\}\gamma_1 + \min\{K_3^*, [x-K_1]^+\}\gamma_3\\
&\quad + [x - K_1 - K_3^*]^+\gamma_2,
\end{aligned}$$

where $K_1 = LD(H-1) + (L-1)(D-1)(H-2)$, $K_2^f = (L-1)D + (L-2)(D-1)$, $K_3^f = 2(D-1)$, $K_2^g = L + D - 2$, $K_3^g = 2(L-1)(D-1)$, *and function* $e(L) = 1$ *when* L *is even,* $e(L) = 0$ *otherwise.*

Proof outline. Proof of Theorem 1 can be found in Appendix. C. For the single- or double-monitor cases ($m = 1$ or 2), considering that type-1, 2 and 3 edges are the least expensive edges in the complete graph K_n in terms of switching cost, the main idea of design (and the proof of Theorem 1) is to construct a traversing route with the most number of least expensive edges (type-1 or type-2 depending on quantity $\beta_S - \frac{2-\sqrt{2}}{2}\beta_A\epsilon$), and as few necessary longer edges as possible. The optimal strategy for these two cases, and their accompanied minimum switching costs Γ^1_{min} and Γ^2_{min}, are presented in Lemma 2 and Lemma 4, respectively. Then for $m > 2$, the switching cost can be upper bounded by evenly dividing the low-cost routes of $m = 1$ or $m = 2$, and then removing the induced dividing edges. As a constructive proof, it also sketches the proposed deterministic strategy f_S^m.

Simulation setting. Configuration of numerical simulations is enumerated in Table 3. Note that the width of the spectrum block $\mathcal{S}$ does not have any unit (similarly for monitoring power parameter δ). We eliminate the unit, instead of plugging in parameters of real-world hardware *e.g.*, [7], [14], because there will be a (mere) change of constant when different units are applied, which is insignificant (and more confusing) in validating the efficiency of the proposed deterministic strategies.

Discussion. The proposed strategy (black bars on the far left in Fig. 5, labeled as f_S) is compared with the best solutions found by genetic algorithm (red-toned bars in Fig. 5.a and b, labeled as GA), which is a commonly used heuristic for MTSP, and those found by a greedy-based random search (blue-toned bars in Fig. 5.c and d, labeled as RS). in particular, GA [31] is chosen as a benchmark due to its capability of finding near-optimal solutions for MTSP with a large search space within a short running time [32], in comparison with other heuristics, such as the ant colony algorithm (ACO) [33], and costly exact algorithms, such as the brute-force search. The proposed f_S achieves a very close switching cost to that of the best solution/strategy output by GA after more than 10,000 iterations, under different switching cost coefficient settings. In fact, when m is small, the best solution provided by GA bears great resemblance to the proposed strategy, in terms of traversed edge types, as well as the breaking points (*i.e.* the way to divide the optimal path of $m = 1$). Similar observations can be obtained in the comparisons with random search, despite that the optimal solution can not be easily found by the random search due to its greedy nature.

4.2 Detection time of the deterministic strategy f_S

Though deterministic strategy f_S proposed in Theorem 1 proves to be a good strategy in terms of the coverage time ($T_S^m = \lceil \frac{|V|}{m} \rceil$) and switching cost, its detection performance is not satisfying. In fact, it suffers from a 'wandering hole' problem when adversarial spectrum culprits are present.

Detecting a persistent culprits R_p. The expected detection time $\mathbb{E}(\tau_p(f_S))$ of different persistent culprits R_p under strategy f_S are presented in Table 2 (proofs in Appendix E), given that the initial spot of R_p is chosen randomly from the assignment space V. As can be seen, the expected detection time $\mathbb{E}(\tau_p(f_S))$ are bounded above by the coverage time T_S^m.

Detecting an adversary culprit R_a. To maintain a low switching cost in the long run, the deterministic strategy f_S will be repeated (in forward/reverse direction for odd/even sweep-coverage cycles) after $T_S = \lceil \frac{n}{m} \rceil$ time. Consequently, it is possible for culprits with learning capabilities to observe the switching pattern of monitors (*e.g.*, by recording the points and time they encounter the SAS monitors), and even *predict* where monitors will not be (*e.g.*, the coverage hole shown in Fig. 2 right) in the next time step. Then culprits can continue chasing the hole, as if hiding in a 'wandering hole' of the dynamically changing coverage $C_t(f_{S,t})$.

Definition 8. ***(Wandering Hole)*** *Strategy* $\{f_t\}_{t\in\mathcal{T}}$ *suffers from a wandering hole problem, if an adversarial culprit* R_a *can exploit the system indefinitely, i.e.,* $\mathbb{E}\big(\tau_a(f_t)\big) = \infty$.

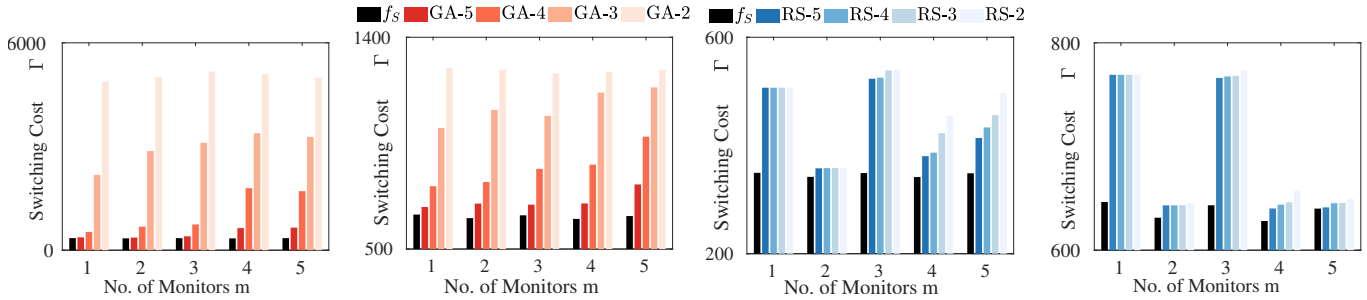
An example. Fig. 6 illustrates a SAS scenario where activities in spectrum block $\mathcal{S}$ are being monitored by $m = 5$ monitors, over the region $\mathcal{A} = [0, 4]^2$. Red dots indicate the assignment points in V, calculated by the Kelvin structure tessellation (Sec. 3.A), while space enclosed by shaded spheres corresponds to the monitoring power of the deployed monitors[12], whose tuning frequency is indicated by the darkness of the shade. The white space outside of these spheres corresponds to spectrum slices and locations where a culprit can exploit without being detected, *i.e.*, a spectrum *hole* in the monitoring coverage C_t. At time $t = 1$, consider an adversarial culprits R_a located at $(3, 3)$ occupying lower frequency portion in the 'hole'. From previous observations, culprit R_a can easily find spectra-location points to exploit in the next time slot $t = 2$, *i.e.*, Void(2) identifies the 'hole' exactly where the probability for these space to be monitored during $t = 2$ equals to zero. Consequently, R_a can safely stay at the current location, and continue occupying the current spectrum slice without being detected. In other words, adversarial culprit R_a can swiftly hide in the 'wandering hole' indefinitely, unless the deterministic deployment strategy f_S changes. In fact, the wandering hole problem exists in any deterministic monitoring strategy. Once the SAS strategy (or more precisely, its probability distribution of visiting) is known by an adversarial culprit R_a, this prior knowledge can be leveraged by R_a to actively dodge monitors, whenever there is a hole in the current coverage, *i.e.*, $X \setminus C_t \neq \phi$.

12. Since we are interested in the closed space $\mathcal{S} \times A$, spheres are trimmed when they intersect with the boundary of $\mathcal{S} \times \mathcal{A}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3032434, IEEE Transactions on Mobile Computing

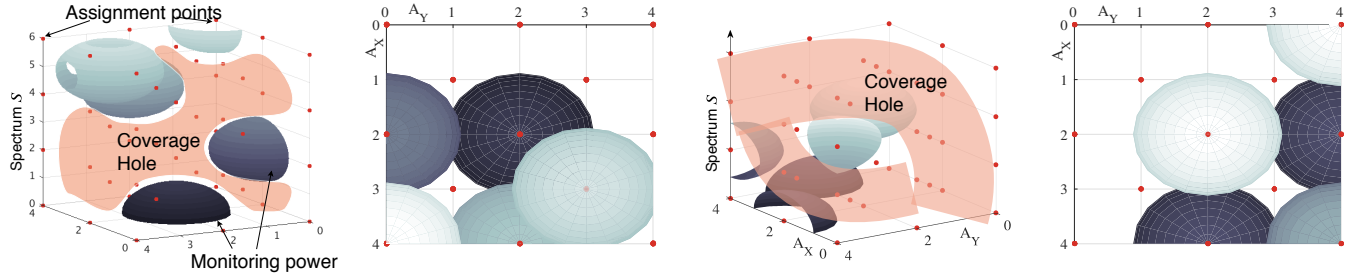
TABLE 3
Simulation Configuration

Parameter	Description	Value	Parameter	Description	Value
n	# of assignment points	394	m	# of monitors	$[1, 5]$
$\mathcal{A}$	geographical region of interest	$[0, 10]^2$	$\lvert\mathcal{S}\rvert$	width of spectrum block	18
ϵ	scaling coefficient in d_{SA}	5	δ	monitoring power of monitors	$\frac{\sqrt{5}}{2}$
(L, D, H)	tessellation coefficients	(5,5,10)	β_S, β_A	switching cost coefficients	$(0, 1)$
GA-k	genetic algorithm with 10^k iterations	$\{2, 3, 4, 5\}$	RS-k	random search algorithm with 10^k iterations	$\{2, 3, 4, 5\}$



(a) GA $\beta_S = 0.1, \beta_A = 0.9$ (b) GA $\beta_S = 0.8, \beta_A = 0.2$ (c) RS $\beta_S = 0.1, \beta_A = 0.9$ (d) RS $\beta_S = 0.8, \beta_A = 0.2$

Fig. 5. The proposed strategy (black on the far left) achieves a lower switching cost, compared to the genetic algorithm solution (red bars in (a-b)) and that the greedy-based random search solution (blue in (c-d)), in different switching cost coefficients (β_A and β_S) settings.



(a) $t = 1$ side view (b) $t = 1$ top view (c) $t = 2$ side view (d) $t = 2$ top view

Fig. 6. Illustration of a *wandering hole* in $\mathcal{S} \times \mathcal{A}$: Five monitors are deployed in region $\mathcal{A} = [0, 4]^2$ with $\delta = \frac{\sqrt{5}}{2}$. Their coverage C_t at each time step t, is the union of the enclosed space of the blue (partial) balls and the box boundaries. The assignment points (V) are shown with red dots. The outter white space of the blue spheres corresponds to the spectrum 'hole' that is 'wandering' (changing) in space X over time.

5 Overcoming the 'Wandering Hole' Problem with Randomized Strategies

The wandering hole problem exposes a defect of deterministic strategies against adversarial culprits R_a. Taking advantage of its prior knowledge, that is, the *difference* in visiting probabilities in the next time step, adversarial culprit R_a is able to determine the spectrum 'hole' to exploit, *i.e.*, Void(t). The sharper the difference, the clearer the boundary of the 'hole', and the larger its chance to dodge monitors in the next time step. For instance, under the deterministic strategy shown in Fig. 6, a culprit R_a located at $a = (3, 2)$ can easily identify Void$(2) \subset \mathcal{S} \times \{a\}$ (the 'bar' on the left in Fig. 7) due to the prominent difference in probability.

Intuition behind the solution. Knowing the root cause of the 'wandering hole' problem, a straight-forward countermeasure is to better protect, or frequently change the SAS strategy, which requires constant effort in the deployment stage, such that obtaining visiting probabilities, or the strategy implementation, are more difficult for culprits. Nonetheless, we can achieve the same goal if we carefully design a SAS strategy, from which no useful 'knowledge' can be derived, even if it is known to the culprit. In other words, fully *randomize* the deployment, such that every assignment point is visited with the same probability in the long run. In this way, there will not be any probability difference, and hence no boundary of spectrum holes for culprits to locate, as illustrated by the uniform grey 'bar' in Fig. 7. In addition, monitors follow no pattern at all when switching to a different assignment point, so that historic

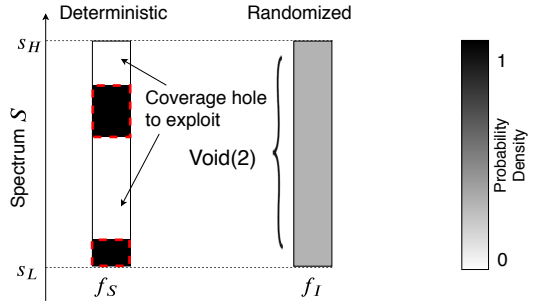


Fig. 7. Root cause of the *wandering hole* problem is the difference in probability density across X in a deterministic strategy f_S. For example, an adversarial culprit located at $(3, 2)$ in Fig. 6 can infer hole Void(2) with clear boundary, which is not the case for a randomized strategy f_I.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3032434, IEEE Transactions on Mobile Computing

record will not add to the knowledge of culprits.

From switching cost to switching capacity. As we shift focus from low-cost sweep-coverage to quick culprit detection, we switch gear from switching cost to switching capacity, which for SAS with dedicated monitors can be regarded as a range determined by a threshold in switching cost. On the other hand, considering that a switching action is actually a change of surrogate monitors, randomized strategies are natural for the crowd-source monitoring scenario. With a centralized coordinator or guaranteed communication among participants, switching between surrogate monitors can be timely coordinated, so the entire assignment space V is contained in the switching capacity of all monitors, which translates to the unlimited switching capacity case, *i.e.*, $\alpha_M = \infty$. In case of distributed control, when change of surrogate monitors needs to be completed with local communication, the switching capacity of monitors will be upper-bounded, *i.e.*, $\alpha_M < \infty$, due to their limited communication ranges.

5.1 Randomized SAS Strategies

We consider two randomized strategies that requires different levels of coordination and switching capacities: the independent I-strategy f_I and the distributed D-strategy f_D.

I-strategy f_I. During each time step, each monitor $M_i \in \mathcal{M}$ switches to point $v_i \in V$ uniformly at random, and independently of others, within its switching capacity α_M. The surveillance process is then equivalent to a composite random walk of $m = |\mathcal{M}|$ walkers, each independently generating a sequence $\{f^m_{t,I}(M)\}_{t\in\mathcal{T}}$, on the monitoring subgraph G_M. When graph G_M is (close to) regular, *i.e.*, the number of assignment points reachable in one switching action is almost the same for every point in V, the uniform transition probability leads to a convergence-guaranteed stationary distribution of visiting probabilities, that is, $\pi_v = \frac{1}{n},\ \forall v \in V$.

D-strategy f_D. Assignment space V is first evenly divided into m disjoint subsets $\{V_i\}_{M_i\in\mathcal{M}}$, such that points in each subset V_i are within monitors' switching capacity α_M, and $\{V_i\}_{M_i\in\mathcal{M}}$ compose a partition of V. During time step t, each monitor $M_i \in \mathcal{M}$ switches to point $f^m_{t,D}(M_i)$, chosen uniformly at random from V_i, which contains $n_m = \lceil \frac{n}{m} \rceil$ assignment points. Thus, the D-strategy can be viewed as m independent single-walker random walks, each on the complete graph K_{n_m}. Moreover, graph K_{n_m} is regular, so the stationary visiting probabilities are also uniform.

Based on this design, we first discuss the basic case of $\alpha_M = \alpha_R = \infty$, *i.e.*, SAS strategy without switching constraint, in this section. As we will show, both the coverage time and detection time of the two randomized strategies: (i) are bounded, indicating their efficacy in both coverage and detection objectives; and (ii) scale as $O(\frac{1}{m})$ with respect to the number of monitors m, revealing their efficiency.

5.2 Coverage Time T^m_I and T^m_D

Because of the unlimited switching capacity of both monitors and culprits, the underlying monitoring subgraph G_M are complete graphs, and both the I- and D-strategy are equivalent to random walks, on K_n and K_{n_m}, respectively. The coverage time T_I and T_D become well-defined r.v.'s that take value in $[1, \infty)$, and their expected value $\mathbb{E}(T_*)$ are referred to as the *cover time* [34].

5.2.1 Coverage Time of the I-strategy T^m_I

For an I-strategy f^m_I carried out by m monitors, the expected coverage time $\mathbb{E}(T^m_I)$ can be bounded by the following theorem.

Theorem 2. ***(Coverage Time of** f^m_I**)*** *For a set of $m = |\mathcal{M}|$ monitors that follow the I-strategy $\{f^m_t\}_{t\in\mathcal{T}}$ in the assignment space V, the expected coverage time is upper bounded by*

$$\mathbb{E}(T^m_I) \le e(n-1)\left[0.562 + 0.768\frac{\mathcal{H}_n}{m}\right], \tag{15}$$

where $n = |V|$ is the number of assignment points in V.

Proof of Theorem 2 can be found in Appendix D. Though not a tight bound, Theorem 2 reveals the scaling law of the expected coverage time with respect to size n of space V, and number of monitors m, that is, $\mathbb{E}(T^m_I) = O(\frac{n\ln n}{m})$.

5.2.2 Coverage Time of the D-strategy T^m_D

The expected coverage time of D-strategy f^m_D can be bounded above with the help of Lemma 5.

Theorem 3. ***(Coverage Time of** f^m_D**)*** *For a set of m monitors following D-strategy f^m_D on the assignment space V of size n, the expected coverage time $\mathbb{E}(T^m_D)$ is upper bounded by*

$$\mathbb{E}(T^m_D) \le n_m\mathcal{H}_{n_m} + \frac{n_m\sqrt{m-1}}{2(n_m-1)}\left[7(n_m)^2 - 11n_m + 2\right]^{\frac{1}{2}}, \tag{16}$$

where $n_m = \lceil \frac{n}{m} \rceil$.

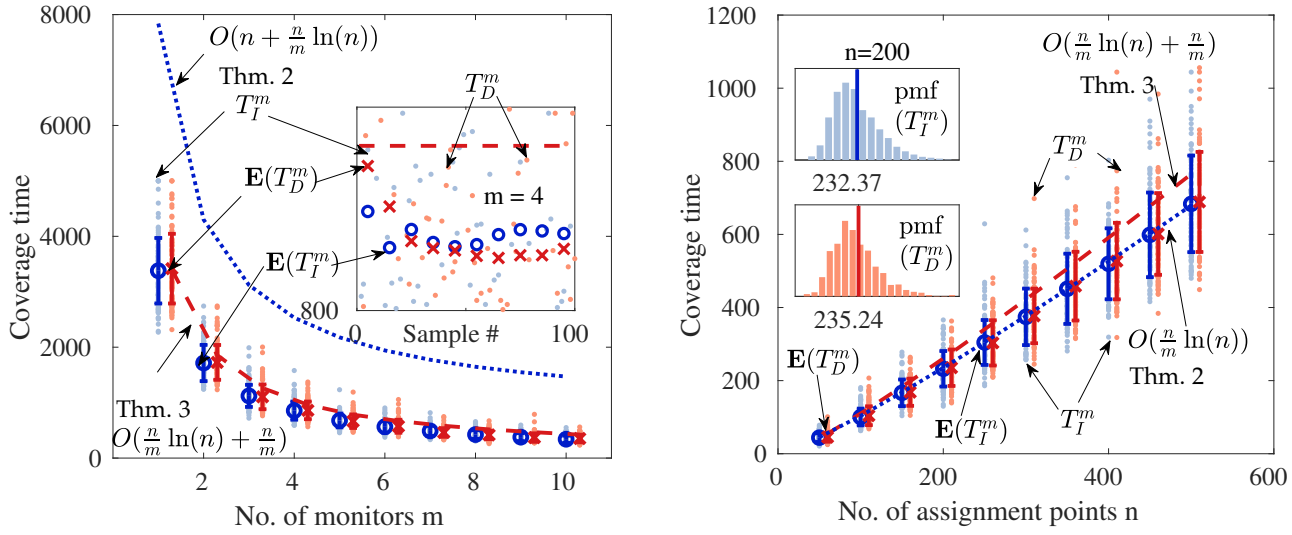
Proof of Theorem 3 can be found in Appendix D.

5.2.3 Simulation Validation

Fig. 8 illustrates the expected coverage time of I-strategy ($\mathbb{E}(T^m_I)$, blue '$\bigcirc$' markers) and D-strategy ($\mathbb{E}(T^m_D)$, red '$\times$' markers) with respect to $m \in [1, 10]$, number of monitors, and $n = |V| \in [50, 500]$, size of the assignment space, respectively. Numerical samples of T_I and T_D are shown in dots, while their mean and standard deviation are shown with markers and bracketed bars. The case of four monitors ($m = 4$) is zoomed in the inner box of Fig. 8 (a), from which it can be seen that even the sliding average of coverage time (round and '$\times$' markers) are upper bounded.

Observations. We have the following observations from the comparison between simulation and bounds: (i) Theorem 3 (red dashed line) is a tight bound on the coverage time of D-strategy when m is small; (ii) Theorem 2 (blue dotted line), though not a tight bound, accurately describes its $O(\frac{n}{m}\ln n)$ scaling behavior; (iii) I-strategy and D-strategy have very close coverage time performances, not only in the mean sense, but also in distribution, as shown in the inner boxes of Fig. 8 (b), which implies that the more demanding I-strategy (in terms of level of coordination and switching capability of monitors) can be safely substituted by the distributed D-strategy, with the same coverage goal guaranteed; iv) both expected coverage times can be described as $O(\frac{n}{m}\ln n)$ (blue dotted line in (b)), indicating that this scaling law can be used to predict the number of monitors

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3032434, IEEE Transactions on Mobile Computing



(a) Scaling as $O(\frac{1}{m})$ over $m = |\mathcal{M}|$ with $n = 500$.

(b) Scaling as $O(n \ln n)$ over $n = |V|$ with $m = 5$.

Fig. 8. Expected coverage time of both I- ($\mathbb{E}(T_I)$) and D-strategy ($\mathbb{E}(T_D)$) are $O(\frac{n}{m}\ln n)$, as predicted by Theorem 2 and Theorem 3, respectively.

needed to reach the coverage goal of a given spectra-location space with a certain resolution.

Discussion. Note the unit of the coverage time is time step, whose length can be evaluated when parameters in Table 3 are determined in a real-world scenario. We do not incorporate specific units in the analysis or simulation, because the focus of this paper is the scaling behavior of strategy performance with respect to the problem size (in this case n) and the number of monitors (m). Compared with the deterministic strategy f_S in Sec. 4, that can achieve a $\frac{n}{m}$ coverage time, randomized strategies seem to be at disadvantage in fulfilling the coverage goal, but as we will show in the following subsection, they are favorable in spectrum culprits detection.

5.3 Bounded Detection Time of Adversarial Culprits

the advantage of adversarial culprits over deterministic strategies is lost, when facing monitors running randomized strategies f_I and f_D, because their prior knowledge (visiting probabilities) is compromised by the uniform probability distribution in f_I and f_D. Consequently, randomized strategies do not suffer from the 'wandering hole' problem, that is, the detection time is bounded.

Theorem 4. ***(No Wandering Hole in Randomized Srategies)*** *Under strategy f_I^m and f_D^m, the expected detection time of an adversarial culprit R_a is upper-bounded, if the detection probability q is lower-bounded by a positive constant $q^* > 0$. Particularly,*

$$\mathbb{E}\big(\tau_a(f_I^m)\big) = \Big[1-(1-\frac{q}{n})^m\Big]^{-1}, \tag{17}$$

$$\mathbb{E}(\tau_a(f_D^m)) = \frac{n}{qm}. \tag{18}$$

Proof of Theorem 4, as well as the closed-form upper bounds, can be found in Appendix F. Technically, it is possible that detecting probability q is minimal, due to the large radius δ in the $q(\delta)$-monitoring power, so the resulting detection time $\mathbb{E}(\tau_a(f_I^m))$ and $\mathbb{E}(\tau_a(f_D^m))$ tend to infinity. However, this can be easily fixed if the radius parameter δ is adjusted in the space-tessellation step, so that q is boosted to an acceptable level. Consequently, we conclude that randomized monitoring strategies (f_I and f_D) do not suffer from the wandering hole problem.

Numerical Simulation. To validate the advantage of randomized strategies against adversarial culprit, stated in Theorem 4, culprit detection is simulated under the same spectra-location space setting as the sweep-coverage validation. Detection time samples are shown as light-blue (I-strategy) and light-red (D-strategy) dots of Fig. 9, which corresponds to the imperfect detection ($q = 0.8 < 1$) case.

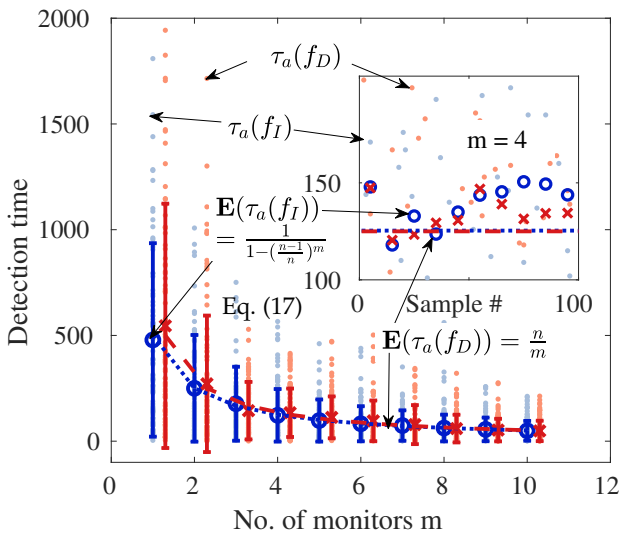
Observations. From the bounds and simulation results, we have the following observations. (i) Not only are the detection time of I-strategy and D-strategy bounded (and hence no 'wandering hole' problem), their expectations can be accurately calculated with Eq. (17) and (18), once the number of monitors m and size of the assignment space n are fixed. (ii) Bounds in Eq. (17) and (18) hold for the imperfect detection case, as shown in Fig. 9. (iii) The detection performance of the I- and D-strategy are fairly close, which indicates that D-strategy can be a good distributed alternative to the I-strategy.

Discussion. The $O(\frac{1}{m})$ scaling behavior in both coverage and detection time, indicates a linear 'speed-up' in SAS performance, when multiple monitors are employed in the randomized strategies. This behavior implies, as same as in the deterministic strategies, increasing the number of monitors (m) is an efficient performance-boosting measure. In addition, the bounds on detection time (or rather, accurate results in Eq. (17) and (18)) add to the predictability of randomized strategies, which can be fairly useful in the design stage of a SAS function, *e.g.*, estimating the needed number of monitors.

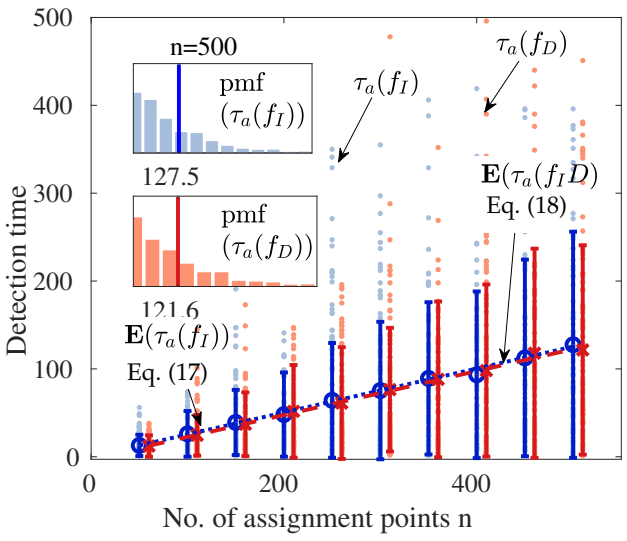
6 SAS under Limited Switching Capacities

Recall in Sec. 3.2, the *switching capacity* α_Y is defined as the maximum distance that a device (monitor or culprit) Y can switch over in one time step. In this section, we consider a SAS process of m independent α_M-monitors (with detection probability $q = 1$) and an α_R-culprit on the assignment

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3032434, IEEE Transactions on Mobile Computing



(a) Scaling as $O(\frac{1}{m})$ over $m = |\mathcal{M}|$ with $n = 500$.



(b) Scaling as $O(n)$ over $n = |V|$ with $m = 5$.

Fig. 9. The expected detection time of an adversarial culprit in the assignment space of size n is $O(\frac{n}{m})$, under both the I and the D-strategy with reliabiltiy $q = 0.8$.

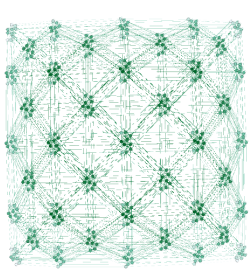
space V. Let $r_{\alpha_M}(v)$ denote the degree of point $v \in V$ in the monitoring subgraph G_M, under constraint α_M, and $r_{\alpha_R}(v)$ denote the degree of v in the exploiting subgraph G_R, under constraint α_R. Then G_M and G_R are both subgraphs of the complete graph K_n, which corresponds to the unlimited case discussed in Sec. 5. When $\alpha_M = \alpha_R$, the culprit and the monitors can be viewed as walking on the same graph, *i.e.*, $G_M = G_R$. However, for cases when monitors are more 'powerful' than the culprit ($\alpha_M > \alpha_R$), edges in G_R are strictly sparser, *i.e.*, $E_R \subset E_M$, and vise versa.

Scenarios. Constrained switching capacity ($\alpha_M < \infty$) applies to the fully-distributed crowd-source SAS scenario, in which a switching is only possible if the two devices are within each other's communication range. Such a capacity limit can also be viewed as a binary quantification of the switching cost, in the sense that it cuts off any edge (v_i, v_j) (in a complete graph), whose associated switching cost $\gamma(v_i, v_j)$ exceeds a threshold.

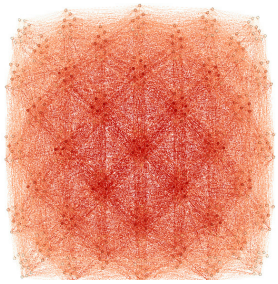
Challenges. The two-step solution introduced in Sec. 3 turned the SAS process into a walk on the composite graph (G_M, G_R), but analysis is still impeded by two challenges: (i) theoretic analysis of random walks on general graphs is difficult, if at all possible, because existing mathematical tools are developed for graphs with special structures, *e.g.*, the complete graph K_n; (ii) monitors and culprits may switch/walk on different graphs, when $\alpha_R \neq \alpha_M$, which was not addressed by existing research on graph walk.

6.1 Solution: Regular Graph Approximation

Observe that assignment points (cell centers in the Kelvin structure) in space V are quite 'structured', as shown in Fig. 3 (right). As a result, subgraph G_M (respectively G_R) (*e.g.*, the weaker monitors v.s. powerful culprit case shown in Fig. 10) that build upon it is also 'structured', in the sense that degrees of most vertices in G_M (G_R) are roughly the same, except for the few near the boundary. Denote $r_M = \frac{1}{n}\sum_{i=1}^{n} r_{\alpha_M}(v_i)$ as the average degree of the monitoring subgraph G_M, and r_R as that of the exploiting subgraph



(a) Monitoring subgraph G_M with $\alpha_M = 5$.



(b) Exploiting subgraph G_R with $\alpha_R = 10$.

Fig. 10. Weaker monitors against a powerful culprit: edge set E_R of the induced exploiting subgraph G_R has much more edges than E_M, edges of the monitoring subgraph G_M.

G_R. We first approximate G_M and G_R as r_M- and r_R-regular graphs (G_{r_M} and G_{r_R}) respectively,[13], on which mathematical tools [34], [35], [36] come in handy.

6.1.1 *Coverage Time* $T^m_{r_M}$

Let $T^m_{r_M}$ denote the coverage time of m independent monitors on r_M-regular graph G_{r_M}, to differentiate from the actual coverage time T^m_I on the original monitoring subgraph G_M. On regular graph G_{r_M}, asymptotic bounds for $\mathbb{E}(T^m_{r_M})$ have been studied by multiple researchers. Among these, Alon *et.al.* [34] proved $\mathbb{E}(T^m_{r_M}) \sim \Theta(\frac{n \ln n}{m})$, as $n \to \infty$; Cooper, Frieze and Radzik [36] provided a similar but more accurate asymptotic result for random regular graphs, when the number of random walkers is not large, *i.e.*, $m = o(\frac{n}{\ln^2 n})$. It is shown in [37] that a uniformly chosen $r-$regular ($r \geq 3$) graph G_r is 'nice' with high probability (tending to one as $n \to \infty$), such that the expected coverage time $T^m_{r_M}$ follows from [36, Thm. 2], that is,

$$\mathbb{E}(T^m_{r_M}) \sim \frac{r_M - 1}{r_M - 2}\frac{n \ln n}{m}. \tag{19}$$

13. This average-degree-based approximation is reasonable, but also introduce a gap when determining SAS performance for both sweep-coverage and culprit detection, which is discussed in Sec. 6.B.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3032434, IEEE Transactions on Mobile Computing

An extension. Under an I-strategy, each monitor switches to any assignment point within its switching capacity uniformly at random, resulting in a uniform stationary distribution $\pi_v = \frac{1}{n}$ over the assignment space V. In other words, each assignment point is visited roughly the same number of times as time proceeds, so we say the spectra-location space X is 'evenly' covered. Nevertheless, if a certain region (subspace of X) needs special attention, *e.g.*, due to higher presence of misbehavior, the probability to switch to a target assignment point can be adjusted, such that a desired (possibly non-uniform) stationary distribution over V can be achieved through well-articulated algorithms, *e.g.*, the Metropolis-Hasting algorithm.

6.1.2 Detection Time $\tau_R(r_R, r_M)$

Unlike coverage time, for which existing research leads to direct solution, there is no proper mathematical tool to directly address the culprit detection problem, in which the monitors and the culprit may walk on *different* graphs, due to their different switching capacity limits. So we address the weaker monitors vs. powerful culprit ($\alpha_M < \alpha_R$) case (and its reverse $\alpha_M > \alpha_R$) in this subsection.

Let $\tau_R(r_R, r_M)$ denote the detection time of a culprit R (walking on the r_R-regular graph G_{r_R}), by m-monitors (walking on the r_M-regular graph G_{r_M}). There are two possible cases:

Case 1. Same switching capacity. Monitors $\mathcal{M}$ and the culprit R have the same switching capacity, such that $r_R = r_M = r$, and $G_{r_R} = G_{r_m} = G_r$, *i.e.*, both monitors and culprit R walk on an r-regular graph G_r. Applying the predictor-and-prey model [36, Theorem 3], the expected detection time of culprit R can be asymptotically bounded, that is,

$$\mathbb{E}(\tau_R(r, r)) \sim \frac{r-1}{r-2} \cdot \frac{n}{m}. \tag{20}$$

Case 2. Different switching capacities. Monitors walk on a different graph from the culprit, *i.e.*, $r_R \neq r_M$ such that $G_{r_R} \neq G_{r_m}$. We obtain the upper bound of the detection time by considering a composite random walk.

Proposition 2. ***(Bounded Detection Time on Regular Graphs)*** *Let $K = \frac{(n-1)!}{(n-m-1)!}$. Under an I-strategy with m monitors, the expected detection time of a culprit on graph (G_{r_M}, G_{r_R}) is upper bounded, i.e.,*

$$\mathbb{E}\left(\tau_R(r_R, r_M)\right) \leq 1 + \frac{K}{n^m}(4K^2 - 1). \tag{21}$$

Proof of Proposition 2 can be found in Appendix G. Proposition 2 holds for every n, which is the number of assignment points in V. However, when n is large, it is not easy to calculate K and n^m in Eq. (21). For this case, we have the following scaling law on the expected detection time.

Corollary 1. ***(Scaling Law of Detection Time)*** *The expected detection time of a culprit for a SAS process under the I-strategy on graph (G_{r_R}, G_{r_M}) satisfies*

$$\mathbb{E}\left(\tau_R(r_R, r_M)\right) = \Theta(\frac{n}{m}). \tag{22}$$

Proof of Corollary 1 can be found in Appendix G. Compared to the SAS scenarios with unlimited switching capacity discussed in Sec. 5, the scaling laws in this regular graph approximation (coverage time Eq. (19), detection time Eq. (20) and Eq. (22)) differ only by a degree-determining constant, which is less than or equal to 2. Consequently, we expect the scaling laws of both performance metrics (over m and n) to remain the same as the unlimited case.

6.2 Gap between (G_M, G_R) and (G_{r_M}, G_{r_R})

For Eq. (19) and Eq. (20) (and Eq. 22) to hold, a regular graph needs to be 'nice' [37, pp. 733]. It is also shown in [37] that a large (n large) r-regular graph G_r randomly selected from the collection of all r-regular graphs $\mathcal{G}_r$, is *almost-Ramanujan* with high probability, that is, the largest eigenvalue $\lambda_0(G_r)$ and the second largest eigenvalue $\lambda_1(G_r)$ of graph G_r's adjacency matrix satisfy

$$\lambda_1(G_r) \leq 2\sqrt{\lambda_0(G_r) - 1} + \epsilon, \tag{23}$$

where the $\lambda_0(G_r) = r$, as G_r is r-regular.

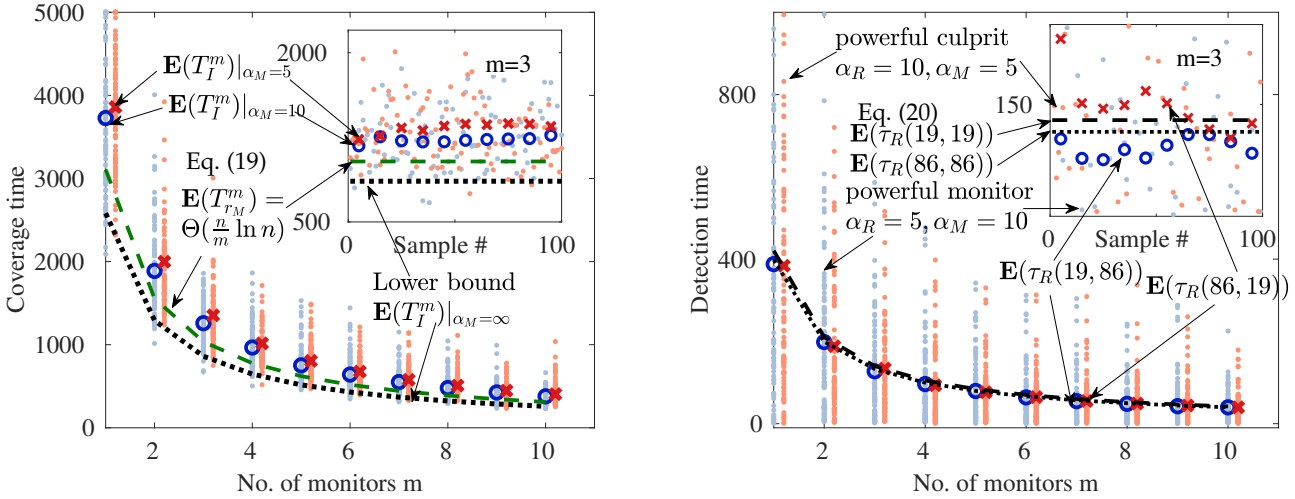
However, Eq. (23) does not necessarily hold for the real exploiting and monitoring subgraphs (G_R, G_M). For instance, the monitoring subgraph G_M presented in Fig. 10 corresponds to $\alpha_M = 5$. This graph G_M has $\lambda_0(G_M) = 18.415$ and $\lambda_1(G_M) = 16.475$, certainly violating the eigenvalue gap criterion in Eq. (23). Nonetheless, a randomly generated graph with the same average degree, G_{r_M} ($r_M = 17$), has $\lambda_0(G_{r_M}) = 17$ and $\lambda_1(G_{r_M}) = 7.633$, satisfying the criterion. The gap in the graph expansion property does not allow direct application of the scaling law (described by Eq. (19) and Eq. (20)) to the composite graph (G_M, G_R), induced by switching capacity limit α_M and α_R, even though (G_M, G_R) have the the same average degree as its regular-graph approximate (G_{r_M}, G_{r_R}) by construction. Therefore, we employ simulation to see if the approximation is valid.

6.3 Numerical Simulation

Simulation Setting. We validate the regular approximation in the same assignment space V, detailed in Table 3. Simulation results (dots) and bounds (dashed and dotted lines) of the coverage time and detection time, under an I-strategy with different switching capacities, are shown in Fig. 11a and 11b, respectively. The powerful monitors case ($\alpha_M = 10$, corresponding to $r_M = 19$, and $\alpha_R = 5$, corresponding to $r_R = 86$) is marked in blue, whose mean is shown by the blue '$\bigcirc$' marker, while the powerful culprit case ($\alpha_M = 5$, $\alpha_R = 10$) is marked in red, whose mean is shown by the red '$\times$' marker. In Fig. 11a, the lower bound of the coverage time (black dotted line) is obtained by setting r_M to ∞ in Eq. (19).

Observations. From the coverage time in Fig. 11a, we observe as we anticipated: (i) The coverage time of a weaker monitor set (red '$\times$' markers, $r_M = 19$) is slightly longer than that of a more powerful monitor set (blue '$\bigcirc$' markers, $\alpha_M = 10$). (ii) The scaling law of the expected coverage time $\mathbb{E}(T_I^m)$ over m is well captured, despite the switching capacity limit. From the detection time in Fig. 11b: (iii) as predicted by Corollary 1, the expected detection time $\mathbb{E}(\tau_R(f_I^m))|_{\alpha_M=5}$ is not lengthened much compared to the strengthened case $\alpha_M = 10$, as opposed to an intuitive anticipation, indicating both the time and range aspects

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3032434, IEEE Transactions on Mobile Computing



(a) Coverage time $\mathbb{E}(T_I^m) = \Theta(\frac{n}{m}\ln n)$. (b) Detection time $\mathbb{E}(\tau_a(f_I^m)) = \Theta(\frac{n}{m})$.

Fig. 11. The expected coverage time and detection time for SAS processes with switching capacity limits (by the regular graph approximation).

of the switching capacity do not impact the expected detection time much. (iv) Both the upper and lower bound of the expected detection time are tight, if not precise ($\mathbb{E}(\tau_R(f_I^M)) \simeq \mathbb{E}(\tau_R(r_R, r_M))$ for $m \in [1, 10]$).

From both figures: (i) Even though the switching capacity of monitors (α_M) and that of the culprit (α_R) differ considerably in value for the two simulation cases, the mean coverage and detection time (round and '$\times$' markers in both Figure 11a and Figure 11b) are pretty close. The reason behind this is similar to what is revealed in Lemma 1, *i.e.*, the more 'mobile' (either monitors or culprits), the more 'visible' to spectrum monitors. (ii) Through (G_M, G_R) are not regular graphs, bounds derived for their regular graph approximation (G_{r_M}, G_{r_R}) (dash and dotted black lines) apply smoothly to both the coverage time, and detection time, in the sense that the scaling laws are well-captured.

Discussion. Comparing the unlimited (Fig. 8 (a) and 9 (a)) with limited (Fig. 11a and 11b) switching capacity cases, the capability limit α_M becomes less influential as the number of monitors m increases, and does not change the scaling behavior over m. The reason behind this is that α_M is sufficiently large so that the quantity $\frac{r_M-1}{r_M-2}$ in Eq. (19) comes close to 1. With extensive simulation, we found that $\mathbb{E}(T_I^m)$ and $\mathbb{E}(\tau_R(f_I^m))$ on the real composite graph (G_M, G_R) actually follow the $\Theta(\frac{n\ln n}{m})$ and $\Theta(\frac{n}{m})$ scaling law described in Eq. (19) and Eq. (20)). We speculate the reason is that both subgraphs (G_M and G_R) are well-connected in degree sense, and the variation in node degree is small so that G_M and G_R are 'regular' enough. On the other hand, this observation makes us wonder whether the requirement of being 'nice' is necessary in achieving the $\Theta(\frac{n\ln n}{m})$ and $\Theta(\frac{n}{m})$ scaling law.

7 Conclusion

In this paper, we study spectrum activity surveillance (SAS) in DSA-enabled systems, particularly deployment strategies of spectrum monitors, for the purpose of sweep-coverage and spectrum culprits detection. We introduce a 3-D model that incorporates spectra, temporal and geographical domains, and captures the locality of different spectrum activities. Under this model, any SAS process can be formulated into a composite walk on a graph generated by efficient space-tessellation, and evaluated with the proposed coverage and detection metrics. As an application of the proposed model, we present a deterministic strategy to achieve quick sweep-coverage with low switching cost, and randomized strategies to achieve quick detection of adversarial culprits. Efficacy of these strategies are theoretically analyzed and validated through simulations. We identify that there are still research efforts to take, before the proposed strategies can be implemented in SAS systems, such as configuring the parameters of the monitor model for real-world SAS monitors, and constructing the SAS graph for systems with complex spectrum ranges, or irregular geographical regions. We hope these results could contribute to the design, analysis, and management of such spectrum-agile systems.

References

[1] O. Holland, H. Bogucka, and A. Medeisis, *Practical Mechanisms Supporting Spectrum Sharing*, pp. 450–. Wiley Telecom, 2015.

[2] M. N. Tehrani, M. Uysal, and H. Yanikomeroglu, "Device-to-device communication in 5g cellular networks: challenges, solutions, and future directions," *IEEE Communications Magazine*, vol. 52, pp. 86–92, May 2014.

[3] Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, and V. K. Bhargava, "A survey on non-orthogonal multiple access for 5g networks: Research challenges and future trends," *IEEE Journal on Selected Areas in Communications*, vol. 35, pp. 2181–2195, Oct 2017.

[4] M. Liyanage, A. Gurtov, and M. Ylianttila, *Leveraging SDN for the 5G Networks*, pp. 440–. Wiley Telecom, 2015.

[5] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Communications Surveys Tutorials*, vol. 17, pp. 358–380, Firstquarter 2015.

[6] D. N. Hatfield, L. Claudy, M. Gorenberg, D. Gurney, G. Lapin, B. Markwalter, G. Mendenhall, P. de Vries, and D. Roberson, "Introduction to interference resolution, enforcement and radio noise," Tech. Rep. FCC 14-31, FCC Technological Advisory Council, Jun. 2014.

[7] A. Nika, Z. Zhang, X. Zhou, B. Y. Zhao, and H. Zheng, "Towards commoditized real-time spectrum monitoring," in *Proceedings of the 1st ACM Workshop on Hot Topics in Wireless*, HotWireless '14, (New York, NY, USA), pp. 25–30, ACM, 2014.

[8] Google, "Google spectrum database: Google earth visualization of available tv white space spectrum." https://www.google.com/get/spectrumdatabase/, 2013. Accessed: 2017-12-18.

[9] Microsoft, "White spaces database." http://whitespaces.microsoftspectrum.com/, 2015. Accessed: 2018-02-11.

[10] M. Höyhtyä, A. Mämmelä, M. Eskola, M. Matinmikko, J. Kalliovaara, J. Ojaniemi, J. Suutala, R. Ekman, R. Bacchus, and D. Roberson, "Spectrum occupancy measurements: A survey and use of interference maps," *IEEE Communications Surveys Tutorials*, vol. 18, pp. 2386–2414, Fourthquarter 2016.

[11] A. M. Voicu, L. Simić, and M. Petrova, "Inter-technology coexistence in a spectrum commons: A case study of wi-fi and lte in the 5-ghz unlicensed band," *IEEE Journal on Selected Areas in Communications*, vol. 34, pp. 3062–3077, Nov 2016.

[12] A. Thapaliya and S. Sengupta, "Understanding the feasibility of machine learning algorithms in a game theoretic environment for dynamic spectrum access," in *2017 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, pp. 1–8, July 2017.

[13] M. J. L. Pan, T. C. Clancy, and R. W. McGwier, "A machine learning approach for dynamic spectrum access radio identification," in *2014 IEEE Global Communications Conference*, pp. 1041–1046, Dec 2014.

[14] D. Pfammatter, D. Giustiniano, and V. Lenders, "A software-defined sensor architecture for large-scale wideband spectrum monitoring," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, IPSN '15, (New York, NY, USA), pp. 71–82, ACM, 2015.

[15] S. Liu, L. J. Greenstein, W. Trappe, and Y. Chen, "Detecting anomalous spectrum usage in dynamic spectrum access networks," *Ad Hoc Networks*, vol. 10, no. 5, pp. 831 – 844, 2012. Special Issue on Cognitive Radio Ad Hoc Networks.

[16] L. Yang, Z. Zhang, B. Y. Zhao, C. Kruegel, and H. Zheng, "Enforcing dynamic spectrum access with spectrum permits," in *Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '12, (New York, NY, USA), pp. 195–204, ACM, 2012.

[17] D.-H. Shin and S. Bagchi, "Optimal monitoring in multi-channel multi-radio wireless mesh networks," in *Proceedings of the Tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '09, (New York, NY, USA), pp. 229–238, ACM, 2009.

[18] X. Jin, J. Sun, R. Zhang, Y. Zhang, and C. Zhang, "Specguard: Spectrum misuse detection in dynamic spectrum access systems," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 172–180, April 2015.

[19] M. Li, D. Yang, J. Lin, M. Li, and J. Tang, "Specwatch: Adversarial spectrum usage monitoring in crns with unknown statistics," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, April 2016.

[20] B. V. den Bergh, D. Giustiniano, H. Cordobés, M. Fuchs, R. Calvo-Palomino, S. Pollin, S. Rajendran, and V. Lenders, "Electrosense: Crowdsourcing spectrum monitoring," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–2, March 2017.

[21] J. Wang, W. Wang, and C. Wang, "Modeling and strategy design for spectrum monitoring over a geographical region," in *2017 IEEE Global Communications Conference: Cognitive Radio and Networks (Globecom2017 CRN)*, (Singapore, Singapore), Dec. 2017.

[22] J. Wang, W. Wang, and C. Wang, "Sas: Modeling and analysis of spectrum activity surveillance in wireless overlay networks," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 2143–2151, April 2019.

[23] Y. Chen, M. Ding, D. Lopez-Perez, J. Li, Z. Lin, and B. Vucetic, "Dynamic reuse of unlicensed spectrum: An inter-working of lte and wifi," *IEEE Wireless Communications*, vol. 24, pp. 52–59, October 2017.

[24] 3GPP, "NR; User Equipment (UE) radio transmission and reception; Part 3: Range 1 and Range 2 Interworking operation with other radios," Technical Specification (TS) 38.101-3, 3rd Generation Partnership Project (3GPP), 01 2019. Version 15.4.0.

[25] "Rohde & schwarz® fsh4/8/13/20 spectrum analyzer operation manual." https://www.rohde-schwarz.com/us/manual/r-s-fsh4-8-13-20-operating-manual-manuals-gb1_78701-29159.html, 2016. Accessed: 2018-2-21.

[26] S. Yoon, L. E. Li, S. C. Liew, R. R. Choudhury, I. Rhee, and K. Tan, "Quicksense: Fast and energy-efficient channel sensing for dynamic spectrum access networks," in *2013 Proceedings IEEE INFOCOM*, pp. 2247–2255, April 2013.

[27] G. Baldini and G. Steri, "A survey of techniques for the identification of mobile phones using the physical fingerprints of the built-in components," *IEEE Communications Surveys Tutorials*, vol. 19, pp. 1761–1789, thirdquarter 2017.

[28] S. W. Thomson, "Lxiii. on the division of space with minimum partitional area," *Philosophical Magazine Series 5*, vol. 24, no. 151, 1887.

[29] D. Weaire and R. Phelan, "A counter-example to kelvin's conjecture on minimal surfaces," *Philosophical Magazine Letters*, vol. 69, no. 2, pp. 107–110, 1994.

[30] M. Jain, J. I. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha, "Practical, real-time, full duplex wireless," in *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, MobiCom '11, (New York, NY, USA), pp. 301–312, ACM, 2011.

[31] K. Sundar and S. Rathinam, "Generalized multiple depot traveling salesmen problem—polyhedral study and exact algorithm," *Computers & Operations Research*, vol. 70, pp. 39 – 55, 2016.

[32] Y. Deng, Y. Liu, and D. Zhou, "An improved genetic algorithm with initial population strategy for symmetric tsp," in *Mathematical Problems in Engineering*, vol. 2015, pp. 3171–3178, 2015.

[33] X. Chen, P. Zhang, G. Du, and F. Li, "Ant colony optimization based memetic algorithm to solve bi-objective multiple traveling salesmen problem for multi-robot systems," *IEEE Access*, vol. 6, pp. 21745–21757, 2018.

[34] N. Alon, C. Avin, M. Koucky, G. Kozma, Z. Lotker, and M. R. Tuttle, "Many random walks are faster than one," *Combinatorics, Probability and Computing*, vol. 20, no. 4, pp. 481–502, 2011.

[35] D. Aldous and J. A. Fill, "Reversible markov chains and random walks on graphs," 2002. Unfinished monograph, recompiled 2014, available at http://www.stat.berkeley.edu/$\sim$aldous/RWG/book.html.

[36] C. Cooper, A. Frieze, and T. Radzik, *Multiple Random Walks and Interacting Particle Systems*, pp. 399–410. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[37] C. Cooper and A. Frieze, "The cover time of random regular graphs," *SIAM Journal on Discrete Mathematics*, vol. 18, no. 4, pp. 728–740, 2005.

[38] D. Bertsimas, K. Natarajan, and C.-P. Teo, "Tight bounds on expected order statistics," *Probab. Eng. Inf. Sci.*, vol. 20, pp. 667–686, Oct. 2006.

Jie Wang (S'15) received her B.S. and M.S. degree in electrical engineering from Tongji University, Shanghai, China, in 2010 and 2013, respectively. She got her Ph.D. in computer engineering at the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC, USA. Her research interests include modeling and performance analysis of wireless networks, data dissemination and edge computing.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3032434, IEEE Transactions on Mobile Computing

Wenye Wang (F'17) received the M.S.E.E and Ph.D degrees in computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 1999 and 2002, respectively. She is a professor with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC, USA. Her research interests include mobile and secure computing, modeling and analysis of wireless networks, network topology, and architecture design. Dr. Wang has been a member of the Association for Computing Machinery since 1998 and a member of the Eta Kappa Nu and Gamma Beta Phi honorary societies since 2001. She was the recipient of the NSF CAREER Award 2006. She was a co-recipient of the 2006 IEEE GLOBECOM Best Student Paper Award - Communication Networks and the 2004 IEEE Conference on Computer Communications and Networks Best Student Paper Award.

Cliff Wang (F'16) graduated from North Carolina State University with a PhD degree in computer engineering in 1996. He currently serves as the computing sciences division chief for the Army Research Office. He is also appointed as an adjunct faculty member of computer science in the College of Engineering at North Carolina State University. Dr. Wang has been carrying out research in the area of computer vision, medical imaging, high speed networks, and most recently information security.

Min Song (F'18) joined Stevens Institute of Technology in July 2018 as Professor and Chair of the Department of Electrical and Computer Engineering. Before joining Stevens, he was the David House Professor, Chair of the Computer Science Department and Professor of Electrical and Computer Engineering at Michigan Tech from 2014 to 2018. He was also the founding director of the Michigan Tech Institute of Computing and Cybersystems. Prior to joining Michigan Tech, Min served as a program director with the National Science Foundation (NSF) from 2010 to 2014. Min's professional career comprises 28 years in academia, government, and industry. Throughout his career, Min has published more than 165 technical papers and held various leadership positions. He served as TPC Co-Chair for many IEEE conferences including ICC and GLOBECOM. He has been serving as a member of the IEEE INFOCOM Steering Committee. He is the recipient of NSF CAREER Award in 2007 and NSF Director's Award in 2012. Min is an IEEE Fellow.